

UNIVERZA V LJUBLJANI
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Gregor Vertovšek

Primerjava ogrodij za razvoj mobilnih aplikacij

DIPLOMSKO DELO

VISOKOŠOLSKI STROKOVNI ŠTUDIJSKI PROGRAM
PRVE STOPNJE: RAČUNALNIŠTVO IN INFORMATIKA

MENTOR: doc. dr. Damjan Vavpotič

Ljubljana 2016

To delo je ponujeno z licenco: *Creative Commons Priznanje avtorstva – Deljenje pod enakimi pogoji, 2.5 Slovenija* (ali novejšo različico). To pomeni, da se tako besedilo, slike, grafike in druge sestavine naloge kot tudi rezultati diplomskega dela lahko prosto distribuirajo, reproducirajo, uporabljajo, priobčujejo javnosti in predelujejo pod pogojem, da se jasno in vidno navede ime avtorja in naslov tega dela ter da se v primeru spremembe, preoblikovanja ali uporabe tega dela v svojem delu lahko distribuira predelava le pod licenco, ki je enaka tej. Podrobnosti licence so dostopne na spletni strani creativecommons.si ali na Inštitutu za intelektualno lastnino, Streliška 1, 1000 Ljubljana.



Izvorna koda diplomskega dela, njeni rezultati in v ta namen razvita programska oprema je ponujena pod licenco GNU General Public License, različica 3 (ali novejša). To pomeni, da se lahko prosto distribuira in/ali predeluje pod navedenimi pogoji. Podrobnosti licence so dostopne na spletni strani: <http://www.gnu.org/licenses/>.

Besedilo je oblikovano z urejevalnikom besedil \LaTeX .

Lektorica: Tanja Tomšič, prof. slov.

Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:

Tematika naloge:

V okviru diplomske naloge predstavite vsaj pet vodilnih ogrodij za razvoj mobilnih aplikacij ter jih med seboj primerjajte. Podlaga za primerjavo naj bo celovit nabor kriterijev, ki bo omogočal, da boste pri primerjavi upoštevali tako podatke proizvajalcev kot tudi podatke s strani razvijalcev, torej uporabnikov ogrodij. V okviru naloge za potrebe primerjave petih izbranih ogrodij pripravite in izvedite anketo s katero boste zbrali mnenja razvijalcev, ki v praksi uporabljajo primerjana ogrodja. Rezultate dela kritično ovrednotite in podajte priporočila za nadaljnje delo.

IZJAVA O AVTORSTVU DIPLOMSKEGA DELA

Spodaj podpisani Gregor Vertovšek sem avtor diplomskega dela z naslovom:

Primerjava ogrodič za razvoj mobilnih aplikacij

S svojim podpisom zagotavljam, da:

- sem diplomsko delo izdelal samostojno pod mentorstvom doc. dr. Damjana Vavpotiča,
- so elektronska oblika diplomskega dela, naslov (slov., angl.), povzetek (slov., angl.) ter ključne besede (slov., angl.) identični s tiskano obliko diplomskega dela,
- soglašam z javno objavo elektronske oblike diplomskega dela na svetovnem spletu preko univerzitetnega spletnega arhiva.

V Ljubljani, 28. januar 2016

Podpis avtorja:

Zahvalil bi se svojemu mentorju, docentu dr. Damjanu Vavpotiču, za: strokovno pomoč, nasvete in podporo pri izdelavi tega diplomskega dela. Prav tako se zahvaljujem tudi svoji družini in moji puncici – vsem, ki so me podpirali in spodbujali v vsem času med mojim študijem.

Kazalo

Povzetek

Abstract

1	Uvod	1
1.1	Pregled poglavij	2
2	Pregled sorodnih evalvacijskih modelov in izbranih mobilnih ogrodi	5
2.1	Model za evalvacijo ogrodi za razvoj mobilnih spletnih aplikacij	6
2.2	Delovanje UI ogrodi za razvoj hibridnih mobilnih aplikacij	9
2.3	Primerjalni kriteriji, osnovani na odprto-kodnih HTML5 UI ogrodi za razvoj na več platformah	12
3	Pregled ogrodi za primerjavo	15
3.1	Splošno	15
3.2	Ionic	17
3.3	JQuery Mobile	18
3.4	Sencha Touch	19
3.5	Xamarin	20
3.6	Framework 7	22

4	Predstavitev evalvacijskega modela	23
4.1	Izbira mobilnih ogrodiv	23
4.2	Izbira kriterijev za primerjavo mobilnih ogrodiv	24
4.3	Sestavljanje uporabniške ankete	28
4.4	Izvedba ankete	30
5	Predstavitev rezultatov modela	33
5.1	Rezultati ankete	33
5.2	Podatki, pridobljeni s strani proizvajalcev in skupnosti	38
5.3	Sinteza rezultatov	49
6	Zaključek	53
	Literatura	55

Slike

5.1	Delež odgovorov za posamezno ogrodje	34
5.2	Poprečja posameznih ogrodi	36
5.3	Uporabniške trditve po skupinah kriterijev	37
5.4	Končne ocene ogrodi	38

SLIKE

Tabele

2.1	Kriteriji modela: Model za evalvacijo ogrodij za razvoj mobilnih spletnih aplikacij[4]	7
2.2	Kriteriji modela: Delovanje UI ogrodij za razvoj hibridnih mobilnih aplikacij[5]	10
2.3	Kriteriji modela: Primerjalni kriteriji, osnovani na odprtokodnih HTML5 UI ogrodjih[6]	13
5.1	Podpora različnih platform	39
5.2	Cene posameznih ogrodij	40
5.3	Podpora za geo-lokacijo	41
5.4	Preprostost sintakse	41
5.5	Podpora vtičnikov	42
5.6	Dokumentacija	43
5.7	Skupnosti	44
5.8	»Native« izgled in občutek	45
5.9	Uporabniška izkušnja	46
5.10	Animacije	47
5.11	Podpora različnih platform	47
5.12	Prikaz podatkov o različicah	48
5.13	Prikaz ocen kriterijev	50
5.14	Prikaz podatkov proizvajalcev	50

Seznam uporabljenih kratic

kratica	angleško	slovensko
HTML5	Hypertext Markup Language 5	Spletni označevalni jezik
CSS3	Cascade Style Sheets	Kaskadne stilske predloge
API	Application programming interface	Vmesnik uporabniškega programa
SDK	Software development kit	Komplet programskih orodij za razvoj
UI	User interface	Uporabniški vmesnik
JSON	JavaScript Object Notation	JavaScript objektna notacija
AJAX	Asynchronous JavaScript and XML	Asinhroni JavaScript in XML
XML	Extensible markup Language	Razširljiv označevalni jezik
I/O	Input/Output	Vhod/Izhod
MVC	Model-View-Controller	Model-pogled-kontrola
SASS	Syntactically Awesome Style Sheets	Sintaktično odlične stilske predloge
GPS	Global Positioning System	Sistem globalnega pozicioniranja

Povzetek

Živimo v času, v katerem je razvoj mobilnih naprav doživel velik razmah, kar iz dneva v dan povzroča vse večje zahteve po razvoju mobilnih aplikacij. Različna podjetja iščejo načine, kako bi lahko svoje storitve in aplikacije čim hitreje in s čim manjšimi stroški prenesla na različne mobilne platforme. Zaradi takšnih zahtev so se v zadnjih letih na trgu začela pojavljati različna mobilna ogrodja, ki omogočajo lažji in hitrejši razvoj mobilnih aplikacij ter hitro prenosljivost na različne mobilne platforme. Velika večina teh ogrodij omogoča razvoj hibridnih mobilnih aplikacij, za katere razvijalci potrebujejo samo znanje spletnih tehnologij. V diplomskem delu se bomo osredotočili na različne lastnosti nekaterih najbolj znanih in popularnih mobilnih ogrodij in na primerjavo teh lastnosti med seboj. Sestavili bomo evalvacijski model, pri njegovi sestavi pa se bomo zgledovali po nekaterih podobnih oziroma sorodnih evalvacijskih modelih. Naš model bo temeljil na mnenjih različnih strokovnjakov in razvijalcev s področja razvoja mobilnih aplikacij. Na podlagi njihovih mnenj in najdenih dejstev želimo razvijalcem dati vpogled v ustreznost in zmogljivost teh mobilnih ogrodij za razreševanje določenih problemov. Poleg tega želimo razložiti, za reševanje katerih težav je najbolje uporabiti katero mobilno ogrodje.

Ključne besede: mobilno ogrodje, android, iOS, spletne tehnologije, model, hibridne mobilne aplikacije, »native« mobilne aplikacije, mobilne spletne aplikacije.

Abstract

We live in a time in which development of mobile devices underwent a major expansion. This expansion is causing a growing demand for mobile application development, with each passing day. Different companies are searching for ways to deliver their services and applications, to different mobile platforms, faster and more cost efficient. In recent years, different mobile frameworks started to emerge on the market, due to such demands. These frameworks enable faster and easier application development and also provide faster deployment on different platforms. Most of these frameworks enable the development of hybrid mobile applications, for which developers only need the knowledge of web technologies. In this thesis, we will focus on the review of different properties, of some well known and popular mobile frameworks and the comparison of these properties. We will create our own evaluation model, based on some of the other similar or related evaluation models. The model will be based on opinions from different developers and experts in the field of mobile development. We want to give the developers, an insight into compliance and performance of these mobile frameworks, based on different opinions and found facts, for specified problems. We also want to pass a recommendation, for which kind of problem solving, which mobile framework is best suited for.

Keywords: mobile framework, android, iOS, web technologies, hybrid mobile applications, »native« mobile applications, mobile web applications..

Poglavje 1

Uvod

V današnjem času, se potrebe po mobilnosti iz dneva v dan povečujejo. Vse več aplikacij se z računalnikov seli na mobilne naprave, kot so: tablice in mobilni telefoni. V zadnjih štirih letih se je mobilnost razvila do te mere, da je s še ene platforme za pregled vsebin prerasla v platformo, ki nam vsak dan pomaga doseči več[1]. S povečevanjem števila uporabnikov mobilnih naprav se povečuje tudi potreba po hitrem razvoju mobilnih aplikacij za različne mobilne platforme. Za doseganje konkurenčnosti morajo podjetja prenosljivost na različne platforme doseči z minimalnimi stroški. Da bi rešili problem prenosljivosti, se je v zadnjih letih na trgu pojavilo kar nekaj mobilnih ogrodij za razvoj hibridnih mobilnih aplikacij.

Ker se ogrodja med sabo razlikujejo, je pomembno vprašanje, ki si ga zastavljajo razvijalci, katero ogrodje je najboljše uporabiti, za določen problem. Na spletu obstaja kar nekaj spletnih strani, ki ponujajo kratek vpogled in prikaz dobrih ter slabih lastnosti določenih mobilnih ogrodij, vendar so te primerjave zelo pomanjkljive, saj prikazujejo le del lastnosti celotnega ogrodja. Drugi problem, ki se pojavlja pri teh primerjavah, je ta, da so to primerjave manj znanih ogrodij oziroma z ogrodji, ki se več ne uporabljajo.

V tem diplomskem delu smo pripravili primerjavo mobilnih ogrodij, ki so trenutno najbolj popularna in uporabljana med razvijalci[2]. Želeli smo

dopolniti razlage o pomanjkljivostih ostalih modelov s tem, da smo želeli vključiti mnenja različnih razvijalcev, ki imajo izkušnje s posameznimi ogrodji. V okviru našega diplomskega dela smo identificirali ključne lastnosti mobilnih ogrodij, na podlagi katerih se razvijalci v praksi odločajo za uporabo mobilnega ogrodja. Nato smo pripravili evalvacijski model, ki združuje vse ključne lastnosti mobilnih ogrodij, te lastnosti pa so podrobneje predstavljene v poglavju 4.2.

Naš cilj je razvijalcem ponuditi orodje, ki jim bo pomagalo pri izbiri najbolj ustreznega ogrodja glede na njihove potrebe. Poleg tega jim želimo dati vpogled v lastnosti nekaterih bolj popularnih mobilnih ogrodij ter jim podati priporočila, za katero ogrodje naj se odločijo, ko razvijajo določen tip mobilne aplikacije.

1.1 Pregled poglavij

Naše diplomsko delo je razdeljeno na: 4 poglavja, uvod in zaključek. V prvem poglavju smo naredili kratek pregled podobnih modelov in sorodnih člankov, objavljenih v znanstvenih zbornikih in znanstvenih revijah. Opređeljene bodo njihove dobre in slabe lastnosti, ki so bile izhodišče za pripravo našega evalvacijskega modela. Drugo poglavje je namenjeno kratkemu pregledu različnih tipov aplikacij v splošnem ter kratkemu pregledu mobilnih ogrodij, ki smo jih izbrali za primerjavo.

V tretjem poglavju je predstavljen naš model, ki predstavlja glavni del našega diplomskega dela. V tem poglavju prikazujemo, zakaj smo izbrali v drugem poglavju omenjena ogrodja, na podlagi katerih kriterijev smo jih primerjali ter s pomočjo katerih orodij smo pridobili podatke za primerjavo mobilnih ogrodij.

V četrtem poglavju predstavljamo rezultate svojega modela. Najprej je predstavljen naš vzorec, s katerim smo izvedli primerjave. Nato so prikazana mnenja, ki so nam jih podali razvijalci, in kratka razlaga rezultatov. Drugi

del poglavja predstavlja prikaz podatkov, ki so bili pridobljeni na spletnih straneh proizvajalcev in na straneh spletnih skupnosti ter forumov. Na koncu so podatki povzeti v tabeli, ki razvijalcem omogoča hiter pregled lastnosti mobilnih ogrodij ter jim olajša izbiro najbolj primerne ograda

Poglavje 2

Pregled sorodnih evalvacijskih modelov in izbranih mobilnih ogrodij

V tem poglavju smo pripravili pregled nekaterih ključnih sorodnih modelov in člankov, ki so nam služili kot osnova za naš evalvacijski model. V uvodu smo napovedali, da na spletu obstaja kar nekaj spletnih strani[25] in dokumentov[24], ki ponujajo vpogled v lastnosti mobilnih ogrodij. Težava, ki se pojavlja pri teh virih, je, da ne nudijo celovitega vpogleda v lastnosti mobilnih ogrodij, ampak ponavadi izpostavijo le izbrane tehnične lastnosti in se osredotočajo na podroben opis le-teh[16]. Nekatere primerjave nudijo le kratek opis posameznih mobilnih ogrodij[3], kar pa je občutno premalo informacij za odločanje, katero mobilno ogrodje izbrati. V nadaljevanju bomo na kratko pregledali in opisali nekatere evalvacijske modele in bomo poskusili izpostaviti njihove prednosti in slabosti.

2.1 Model za evalvacijo ogrodij za razvoj mobilnih spletnih aplikacij

Prvi pregledani članek so ustvarili raziskovalci: **Henning Heitkotter, Tim A. Majchrzak, Benjamin Ruland in Till Weber** iz Univerze v Munstru[4]. V članku so med sabo primerjali ogrodja **jQuery mobile, The-M-Project, Sencha Touch in GWT+mgwt**. Članek je razdeljen na 5 delov in opisuje zgradbo modela ter predstavitev rezultatov primerjave ogrodij.

Evalvacijski model, ki so ga sestavili raziskovalci, je eden izmed prvih modelov, kjer so bili za primerjavo ogrodij uporabljeni kriteriji, prikazani v tabeli 2.1. Raziskovalci so kriterije razdelili na dva dela, in sicer na uporabniški in razvijalski del, s katerima so želeli razkriti pogled na izbiro ogrodij. Cilj razdelitve kriterijev pa je bil: izbrati najbolj optimalno mobilno ogrodje glede na zahteve razvijalcev in tudi uporabnikov. Model so raziskovalci uporabili v dveh korakih. Prvi korak je vključeval zbiranje mnenj s pomočjo vprašalnikov, drugi korak pa razvoj prototipov aplikacij s pomočjo pridobljenih mnenj in informacij. Na koncu so raziskovalci rezultate modela povzeli v tabeli, kjer je bil vsak kriterij ocenjen z ocenami od 1 (zelo dobro) do 6 (zelo slabo). Poleg tabele so raziskovalci razložili še opis delovanja prototipov in težav, ki so se pojavljale pri izvajanju prototipov.

Dobre lastnosti, ki smo jih zasledili v celotnem članku:

- **Uporaba kriterijev.** - večina drugih modelov ne uporablja jasno opredeljenih kriterijev. Vsi modeli, ki so opisani v tem dokumentu, imajo jasno opredeljene kriterije, ki pa se v vsakem modelu razlikujejo. Razlikujejo pa se predvsem zato, ker so raziskovalci v svoje modele vključili tiste kriterije, ki so se jim zdeli pomembni za primerjavo.
- **Razdelitev na uporabniški in razvijalski pogled.** - model prikaže, katere lastnosti so pomembne z vidika razvijalca ali pa z vidika uporabnika.

Kriterij	Opis
Licence in Stroški	Stroški povezani s pridobivanjem ogrodja in licenc.
Dolgoročna izvedljivost	Ali se bo ogrodje vzdrževalo in posodabljal na dolgi rok.
Dokumentacija in podpora	Kako dobre informacije zagotavljata dokumentacija in skupnost.
Učni uspeh	Koliko časa potrebuje razvijalec, da se nauči uporabljati mobilno ogrodje.
Razvojni napor	Čas, potreben za razvoj mobilnih aplikacij, ki je odvisen od razumljive sintakse, ponovne uporabnosti kode in dobre podpore orodij.
Razširljivost	Ali ogrodje podpira razširitev svojih funkcionalnosti ter uporabo vtičnikov.
Vzdrževanje	Ogrodje mora zagotavljati podporo vzdrževanju mobilnih aplikacij v prihodnosti.
Elementi Uporabniškega Vmesnika	Kako dobro so zasnovani in optimizirani UI elementi, ki jih ponuja ogrodje.
»Native« izgled in občutek	Ali aplikacija, zgrajena z ogrodjem, deluje enako kot aplikacija, zgrajena z »native« orodji.
Nalagalni čas	Kako hitro se mobilna aplikacija naloži, glede na nestabilne spletne povezave.
Zmogljivosti izvajalnega časa	Kako visoke so zmogljivosti aplikacije v izvajalnem času z vidika uporabnika.

Tabela 2.1: Kriteriji modela: Model za evalvacijo ogrodij za razvoj mobilnih spletnih aplikacij[4]

- **Prikaz močnih in šibkih točk ogrodij.** - Model prikazuje, katere močne oziroma šibke točke ima posamezno ogrodje. Na podlagi tega omogoča, da se razvijalci lažje odločijo za izbiro posameznega mobilnega ogrodja.
- **Priporočila razvijalcem** - Model priporoča razvijalcem, katero ogrodje uporabiti v neki situaciji.
- **Pregled podobnih del** - Model opisuje, kako se razlikuje od ostalih modelov ter v čem te modele dopolnjuje.
- **Dva recenzenta.** - Ogrodja sta med seboj primerjala dva recenzenta, kar lahko pomaga izboljšati model še posebej, če sta strokovnjaka na tem področju. Če pa si mnenji recenzentov glede kriterijev nasprotujeta, lahko kriterije dodatno raziščeta. Če ne najdeta konkretnih argumentov, ki bi podpirali sporne kriterije, jih lahko izločita.

Slabe lastnosti, ki smo jih zasledili v celotnem članku:

- **Ocene osnovane na lastni izkušnji.** - Raziskovalci so ocene za posamezno lastnost osnovali na svojih izkušnjah, ki pa so lahko tudi subjektivne in je lahko zaradi tega model slabše zasnovan.
- **Ni vključenega mnenja strokovnjakov.** - Model ne vključuje mnenj strokovnjakov za posamezna mobilna ogrodja.
- **Pomanjkljivo prikazani rezultati prototipov.** - Pri prikazu rezultatov ni grafičnega prikaza rezultatov, ki so jih raziskovalci pridobili s pomočjo prototipov. Poleg tega opisi ne ponujajo dovolj informacij o sami izvedbi testov s pomočjo prototipov.

2.2 Delovanje UI ogrodij za razvoj hibridnih mobilnih aplikacij

Drugi članek, ki smo si ga pogledali, sta ustvarila raziskovalca **Radek Vala in Roman Jasek** iz Univerze Tomasa Bate v Zlinu[5]. V njem med sabo primerjata ogrodja **Emy, ChocolateChip-UI, Intel App FW, jQTouch, jQuery mobile, PhoneJS in TopCoat**. Članek je razdeljen na 5 delov ter opisuje zgradbo modela in testiranje ogrodij s pomočjo preprostih mobilnih aplikacij.

Evalvacijski model, ki sta ga sestavila raziskovalca, prikazuje izbiranje kriterijev, prikazanih v tabeli 2.2 in urejanje teh kriterijev po pomembnosti. V tem modelu kriteriji niso razdeljeni na uporabniški in razvijalski del, ampak sta ta dva dela združena v enem delu. Kriteriji modela, prikazani v tabeli 2.2, so razdeljeni po pomembnosti, kar pomeni, da pomembnejši kriteriji igrajo večjo vlogo pri izbiri mobilnega ogrodja. Ocene pomembnosti kriterijev tega modela so določili štirje strokovnjaki. Model sta raziskovalca uporabila za testiranje zmogljivosti posameznih mobilnih ogrodij. S pomočjo vsakega izmed mobilnih ogrodij, ki sta jih primerjala med sabo, sta razvila enotno mobilno aplikacijo za testiranje lastnosti, kot so: nalagalni čas, gladko drsenje in gladko premikanje med stranmi. Testiranje teh lastnosti je potekalo v treh korakih:

1. **Metodologija merjenja.** - Opis metodologije merjenja in uporaba tehnologij za merjenje.
2. **Opis rezultatov.** - Opis in primerjava rezultatov lastnosti mobilnih ogrodij.
3. **Prikaz rezultatov.** - Grafična predstavitev rezultatov meritev posameznih mobilnih ogrodij.

Na koncu sta raziskovalca podala rezultate lastnosti v obliki grafov in tabel, ki so prikazovali rezultate testiranj in rezultate samega modela.

Kriterij	Opis
Primernost za razvoj mobilnih aplikacij	Ocenjuje, kako primerno je ogrodje za razvoj mobilnih aplikacij oziroma ali ogrodje vsebuje pogoste UI elemente za mobilne platforme.
Primernost za razvoj namiznih aplikacij	Ocenjuje ali ogrodje vsebuje UI elemente, pogosto najdene v namiznih aplikacijah.
Aktualnost	Ocenjuje, kako pogoste so mesečne posodobitve mobilnih ogrodij ter datumi zadnjih posodobitev.
Licence	Ocenjuje, ali lahko ogrodja uporabljamo v komercialne namene brez omejitev oziroma ali je potrebno kupiti kakšno licenco.
Dokumentacija	Ocenjuje razpoložljivost in kvaliteto dokumentacije, kar direktno vpliva na učno krivuljo pri novih tehnologijah.
Velikost	Ocenjuje najmanjšo velikost izvirne kode, ki jo ustvari ogrodje. Potrebna je za uvoz v projekt mobilne aplikacije.
»Native« izgled	Opisuje, ali se aplikacija, zgrajena z ogrodjem, obnaša podobno kot »native« aplikacija in s tem izboljšuje uporabniško izkušnjo.
Skupnost	Ocenjuje velikost in kvaliteto skupnosti, ki podpira mobilno ogrodje. Ta kriterij označuje nadaljnji razvoj mobilnega ogrodja.

Tabela 2.2: Kriteriji modela: Delovanje UI ogrodij za razvoj hibridnih mobilnih aplikacij[5]

Dobre lastnosti, ki smo jih zasledili v celotnem članku:

- **Hierarhičnost/razvrščanje kriterijev.** - Model razvršča kriterije glede na njihovo pomembnost pri odločanju in primerjavah.
- **Grafičen prikaz rezultatov.** - Aplikacija, ki je bila zasnovana na podlagi modela, prikazuje rezultate primerjav v obliki grafov in v tabeli.
- **Izvedba zmogljivostnih testov.** - Prikaz zmogljivosti aplikacij, zgrajenih s pomočjo posameznih mobilnih ogrodij, ter njihova primerjava v obliki grafov.
- **Teže kriterijev, ocenjene s strani strokovnjakov** - Štirje strokovnjaki so ocenili, kako pomemben je posamezni kriterij pri primerjavi posameznih ogrodij.

Slabe lastnosti, ki smo jih zasledili v celotnem članku:

- **Kratka primerjava.** - Primerjava ogrodij po kriterijih je zelo kratka in ne daje dovolj informacij.
- **Nerazumljiva primerjava.** - Primerjava ogrodij po kriterijih ni dovolj nazorno razložena z vidika razvijalca in ne koristi preveč pri odločanju, katero ogrodje uporabiti.
- **Ni podanih priporočil razvijalcem.** - Model se osredotoča samo na prikaz rezultatov primerjave in ne podaja priporočil razvijalcem.
- **Opis rezultatov je kratek povzetek.** - Opis rezultatov primerjav je le kratek povzetek o tem, kaj je bilo raziskovano v posameznem delu.

2.3 Primerjalni kriteriji, osnovani na odprtokodnih HTML5 UI ogrodjih za razvoj na več platformah

Tretji članek, ki si ga bomo pogledali, so ustvarili raziskovalci: **Hyo-jung Sohn, Min-gyu Lee, Baek-min Seong in Jong-Bae Kim** iz Univerze v Seoulu[6]. V njem so raziskovalci med seboj primerjali ogrodja **Ionic, Intel's App Framework, OnsenUI in The-M-Project**. Članek je zgrajen iz 4 delov in opisuje izbiro podatkov, zgradbo modela ter primerjavo ogrodij in rezultate primerjave.

Evalvacijski model, ki so ga sestavili raziskovalci, prikazuje izpeljavo kriterijev za primerjavo mobilnih ogrodij, prikazanih v tabeli 2.3. Raziskovalci so kriterije izpeljali iz evalvacijskega standarda za mobilna ogrodja. Standard je že v osnovi zgrajen tako, da deli kriterije na uporabniški in razvijalski del. Enako kot v poglavju 2.2 so razvijalci kriterije iz uporabniškega in razvijalskega dela združili v en skupen pogled. Model so raziskovalci uporabili ob iskanju podatkov in informacij o posameznih mobilnih ogrodjih. Podatke so poiskali na uradnih spletnih straneh za mobilna ogrodja in na zunanjih razvijalskih forumih in skupnostih. Rezultati modela so bili prikazani v tabeli, kjer je bil vsak kriterij za vsako posamezno ogrodje ocenjen z ocenami od 1 (zelo slabo) do 5 (zelo dobro). Ocenitvene kriterije so raziskovalci podkrepili še z razlago njihovih ocen za vsako ogrodje.

Dobre lastnosti, ki smo jih zasledili v celotnem članku:

- **Izbira kriterijev po evalvacijskem standardu.** - Raziskovalci so izbrali kriterije glede na opisanike v razvijalskem standardu.
- **Rezultati, predstavljeni v tabeli.** - Rezultati primerjav so predstavljeni v tabeli, kar nudi enoten vpogled v vse lastnosti ogrodij
- **Zbiranje podatkov v skupnostih.** - Raziskovalci so zbirali podatke

Kriterij	Opis
Enotnost UI	Ocenjuje, ali elementi UI dajejo enoten občutek aplikaciji, ne glede na mobilno platformo oziroma ali je ta optimalno izražen.
Uporabniška izkušnja	Ocenjuje, kako gladke so uporabniške interakcije in animacije in kako se aplikacija občuti na dotik.
Nalagalni čas	Ocenjuje čas, ki je potreben za prvi zagon aplikacije v primerjavi z »native« aplikacijo.
Zmogljivosti	Ocenjuje, kako dobro deluje aplikacija, kadar koli in kjer koli uporabnik želi.
Licence	Ocenjuje, kateri tipi licenc so prilagojeni za določeno ogrodje. (Odprte in zaprte licence).
Skupnosti	Ocenjuje, kako močne so skupnosti za posamezno mobilno ogrodje in kakšno podporo nudijo.
Dokumentacija	Ocenjuje stopnjo, na kateri se nahaja dokumentacija ter njena podpora. Pomaga nam zmanjšati razvijalski napor.
Zahtevnost učenja	Ocenjuje, ali ogrodje izpolnjuje obstoječe razvijalske standarde oziroma je modularno, kar lahko pomaga pri lažjem razumevanju ogrodja.
Več platform	Ocenjuje, ali ogrodje podpira več mobilnih platform, kar lahko zmanjša stroške in razvijalski napor.

Tabela 2.3: Kriteriji modela: Primerjalni kriteriji, osnovani na odprto-kodnih HTML5 UI ogrodjih[6]

o posameznem ogrodju, znotraj posameznih skupnosti.

Slabe lastnosti, ki smo jih zasledili v celotnem članku:

- **Pomanjkanje opisa mobilnih ogrodij** - Za vsako izmed ogrodij manjka kratek opis.
- **Združitev kriterijev.** - Model ne prikazuje lastnosti, pomembnih z vidika razvijalca in uporabnika.
- **Ne vključuje mnenj razvijalcev.** - Za razliko od modelov v poglavjih 2.1 in 2.2 ob uporabi tega modela ni bilo vključenih mnenj razvijalcev in strokovnjakov.
- **Prototip Aplikacije.** - Raziskovalci bi lahko sestavili prototipe aplikacij za posamezna ogrodja in ob njih izvedli testiranja lastnosti.

Poglavje 3

Pregled ogrodij za primerjavo

V tem poglavju so na kratko predstavljena ogrodja za razvoj mobilnih aplikacij, med katerimi smo izvedli primerjave. Poleg tega je na kratko predstavljen koncept mobilnih spletnih aplikacij, hibridnih mobilnih aplikacij in »native« mobilnih aplikacij ter njihove lastnosti.

3.1 Splošno

V splošnem se mobilne aplikacije, razvite s pomočjo različnih orodij za razvoj, delijo na tri skupine:

- Mobilne spletne aplikacije.
- »Native« mobilne aplikacije.
- Hibridne mobilne aplikacije.

Mobilne spletne aplikacije so v osnovi spletne strani, ki na občutek delujejo kot spletne aplikacije, vendar lahko prek mobilnih brskalnikov neposredno dostopajo do strojne opreme mobilnih naprav. Zgrajene so s pomočjo spletnih tehnologij, HTML5, CSS3 in JavaScript ter so namenjene uporabi na mobilnih napravah, lahko pa so tudi optimizirane različice že obstoječih

»native« mobilnih aplikacij. Za razliko od standardnih »native« mobilnih aplikacij se ta vrsta ne naloži direktno na napravo, ampak do njih dostopamo prek spletnega brskalnika. Ta skupina ni vedno najbolj optimalna izbira za prenos na različne mobilne platforme, kar je predvsem odvisno od zahtev posameznih mobilnih platform[4].

»Native« **mobilne aplikacije** so v osnovi programi, ki se izvajajo na mobilnih napravah in tako neposredno dostopajo do strojne opreme mobilnih naprav. Zgrajene so s pomočjo programskih jezikov, ki so specifični za posamezne mobilne platforme (npr. Android za razvoj uporablja programski jezik Java, iOS uporablja Objective-C itd.). Ta vrsta mobilnih aplikacij je naložena neposredno na napravo in se tam tudi izvaja kot samostojen program. Največjo težavo pri tej skupini mobilnih aplikacij predstavlja cenovno drag in težaven razvoj. Razvijalci morajo za razvoj mobilnih aplikacij, ki so namenjene za delovanje na več mobilnih platformah, imeti znanje oziroma izkušnje z uporabo programskih jezikov, ki so specifični za vsako mobilno platformo posebej [27].

Hibridne mobilne aplikacije so mobilne aplikacije, ki se izvajajo v »native« vsebovalniku in se s pomočjo internega brskalnika (WebView) naprave prikazujejo kot lokalno shranjene spletne strani. Združujejo koncepte spletnih in »native« mobilnih aplikacij. Zgrajene so s pomočjo spletnih tehnologij HTML5, CSS3 in JavaScript, lahko jim dodajamo tudi »native« kodo, ki je specifična za posamezne mobilne platforme. Poleg tega se lahko prek raznih API-jev povežemo z »native« funkcijami mobilnih naprav. Ta skupina predstavlja preprost, hiter in cenovno ugoden razvoj mobilnih aplikacij, saj razvijalci v veliki meri potrebujejo le znanje spletnih tehnologij. Hibridne aplikacije se enako kot »native« aplikacije izvajajo neposredno na mobilni napravi, vendar pa jih, glede na tip izvajanja na napravi, ločimo na dve podskupini[28]:

- **Aplikacije, ki se izvajajo v internem brskalniku**, so hibridne aplikacije, ki so zgrajene s pomočjo spletnih tehnologij in se na napravi

izvajajo s pomočjo internega brskalnika (WebView). Interni brskalnik, kodo izvaja znotraj »native« vsebovalnika in tako omogoča, da hibridne aplikacije po izgledu in občutku delujejo kot »native« mobilne aplikacije. Tipična mobilna ogrodja, ki delujejo s pomočjo internega brskalnika, so: Ionic, jQuery mobile, Framework 7 in Sencha Touch.

- **Aplikacije, ki se prevedejo v »native« kodo mobilne platforme,** so hibridne aplikacije, ki so zgrajene s pomočjo enega »native« programskega jezika. Namesto poznavanja programskih jezikov za vsako platformo posebej ta tip razvijalcem omogoča, da lahko s pomočjo enega programskega jezika razvijajo aplikacije za več mobilnih platform. Primer: s pomočjo jezika C# lahko poleg razvoja Windows Phone aplikacij razvijamo tudi Android in iOS aplikacije. To lahko razvijalcem prihrani veliko časa, ki bi ga porabili za učenje ostalih »native« programskih jezikov. Tipičen primer ogrodja, ki prevede aplikacijo v »native« kodo, je ogrodje Xamarin.

3.2 Ionic

Ionic je odprto kodno SDK mobilno ogrodje, ki ga je leta 2013 razvilo podjetje Drifty Co. Podjetje se je, upoštevajoč mnenja strank, odločilo razviti svojo lastno mobilno ogrodje, ki bo osredotočeno na zmogljivost in bo zgrajeno s pomočjo modernih spletnih standardov, kot so: HTML5, CSS3 in JavaScript. Ogrodje je zgrajeno na tehnologijah Angular.js in Apache Cordova, ki omogočata hitre interakcije in prenosljivost ogrodja na različne mobilne platforme. Poleg omenjenih tehnologij ogrodje Ionic podpira tudi tehnologijo SASS, s pomočjo katere lahko razvijalci še dodatno izboljšajo uporabniško izkušnjo.

Značilnosti ogrodja Ionic so:

- Gre za odprto-kodno ogrodje in je prosto za uporabo (uporaba MIT-

licence).

- Zgrajeno je ob knjižnici Angular.js, ki dodatno izboljša delovanje ogrodja.
- Ima podporo za platformi iOS in Android (trenutno samo na napravah, ki podpirajo WebKit).
- Je zelo optimizirano za uporabo na napravah, ki uporabljajo zaslon na dotik.
- Ima dobro podporo ukazni vrstici s svojimi lastnimi ukazi (Cordova).
- Opcijska SASS podpora za boljšo uporabniško izkušnjo.
- Podpira tehnologije Cordova, PhoneGap in Trigger.io.
- Ima več kot 500 različnih predlog, narejenih po meri in licenciranih z MIT licenco.

Ogrodje Ionic je osredotočeno predvsem na izgled in občutek ter UI interakcije v mobilnih aplikacijah. Ogrodje poenostavlja front-end del mobilne aplikacije in za svoje delovanje s polnim potencialom potrebuje tehnologijo Angular.js. Poleg tega zagotavlja tudi lastno ukazno vrstico, da lahko razvijalci svoje projekte ustvarijo s preprostimi ukazi. Ukaze, ki jih lahko razvijalci uporabljajo v ukazni vrstici, omogoča ogrodje Cordova, ki poleg tega skrbi tudi za dodajanje vtičnikov v projekte[8]. Trenutna različica ogrodja Ionic je 1.1.1, izdana 5. novembra 2015.

3.3 JQuery Mobile

JQuery Mobile je mobilno ogrodje oziroma bolj natančno JavaScript knjižnica, ki je optimizirana za naprave na dotik. Razvoj ogrodja je osredotočen predvsem na to, da je kompatibilno z različnimi mobilnimi telefoni in tablicami.

Zgrajeno je okrog knjižnice jQuery, kar razvijalcem, ki imajo izkušnje z uporabo te knjižnice, omogoča preprosto uporabo ogrodja. Značilnosti ogrodja jQuery mobile so naslednje:

- Je odprto-kodno in prosto za uporabo.
- Je kompatibilno z vsemi večjimi brskalniki in mobilnimi platformami.
- Zgrajeno okrog jedra knjižnice jQuery.
- Omogoča ustvarjanje lastnih tem po meri.
- Ima omejene odvisnosti in je lahkotno za optimizacijo hitrosti delovanja.
- UI vmesniki so optimizirani na dotik.
- Podpira PhoneGap kot »native« vsebovalnik
- Navigacije so ustvarjene z AJAX klici in omogočajo animirane prehode med stranmi[9].

Ogrodje je razvila ekipa prostovoljcev iz fundacije **the JQuery Foundation**, prva različica ogrodja pa je bila izdana 16. oktobra 2010. Glavni princip, ki se ga ustvarjalci ogrodja držijo, je, da lahko z malo pisanja naredimo več. Ta princip pomeni, da lahko razvijalci ustvarijo eno samo odzivno mobilno aplikacijo in bo le-ta delovala na vseh mobilnih napravah[10]. Zadnja izdana različica ogrodja jQuery mobile je 1.4.5, izdana 31. oktobra 2014.

3.4 Sencha Touch

Sencha Touch je ogrodje, ki je zgrajeno posebej za mobilni splet. Razvijalci ga lahko uporabijo za razvoj uporabniških vmesnikov spletnih aplikacij, ki morajo na občutek delovati kot »native« mobilne aplikacije. Ogrodje v celoti temelji na spletnih standardih, kot so: HTML5, CSS3 in JavaScript[11].

Poleg omenjenih tehnologij ogrodje podpira tudi MVC arhitekturo, zaradi katere je zelo okorno in bolj primerno za razvoj kompleksnejših mobilnih aplikacij [12]. Cilj ogrodja je omogočiti razvijalcem, da hitro in učinkovito razvijejo hibridne mobilne aplikacije za različne mobilne platforme in da te aplikacije nudijo »native« izkušnjo.

Značilnosti ogrodja Sencha Touch so:

- Omogoča visoko razvite performančne »native« UI vmesnike.
- Je namenjeno komercialni uporabi (na voljo pa je tudi prosta različica).
- Omogoča prilagodljive postavitve, animacije in gladka drsenja.
- Podpira uporabo MVC arhitekture.
- Vgrajena je podpora za ustvarjanje lastnih tem.
- Podpira ogrodji Cordova in Phonegap za prenos aplikacij na različne platforme.
- Uporabimo ga lahko za mobilni razvoj in klasičen spletni razvoj.
- Omogoča več kot 50 JavaScript gradnikov (Widgets).

Ogrodje Sencha Touch je ustvarilo podjetje Sencha in je bilo izdano novembra leta 2010, zaradi česar je to eno izmed najstarejših mobilnih ogrodij. Trenutna različica ogrodja je različica 2.4.2, ki je bila izdana 15. junija 2015.

3.5 Xamarin

Xamarin je mobilno ogrodje, ki ga je razvilo in izdalo istoimensko podjetje maja leta 2011. Ogrodje Xamarin za razvoj uporablja programski jezik C# in s tem omogoča razvoj iOS, Android in Windows phone aplikacij, z »native« uporabniškimi vmesniki. Xamarin ogrodje omogoča tudi prenašanje

kode, na več različnih platform. Podjetje Xamarin ponuja več različnih orodij za razvoj mobilnih aplikacij. Tukaj se bomo osredotočili predvsem na ogrodje Xamarin in njegovi knjižnici Xamarin.Android in Xamarin.iOS.

Ti dve knjižnici nam omogočata razvoj »native« aplikacij v programskem jeziku C#, za platformi Android in iOS. Razvijalci lahko že obstoječo kodo prenašajo med obema platformama. Poleg tega lahko ogrodje integriramo s programskim orodjem Visual Studio in s tem razširimo razvoj tudi na platformo Windows Phone. S pomočjo ogrodja lahko integriramo tudi UI kontrole, backend sisteme, oblacne storitve in druge knjižnice direktno v mobilne aplikacije[13]. Trenutna različica knjižnice Xamarin.Android je 6.0, knjižnica Xamarin.iOS pa ima trenutno različico 9.2.

Lastnosti ogrodja Xamarin so:

- Je namenjeno večjim in srednje velikim podjetjem ter v komercialne namene.
- Nudi »native« razvoj mobilnih aplikacij s pomočjo programskega jezika C#.
- Omogoča uporabo obstoječe Java in Objective-C kode v aplikacijah s pomočjo knjižnic Xamarin.Android in Xamarin.iOS.
- Omogoča uporabo »native« uporabniških vmesnikov za najboljšo uporabniško izkušnjo.
- Omogoča dostop do Android in iOS API-jev s pomočjo generatorja vezav.
- Je kompatibilno z večjimi mobilnimi platformami.
- Omogoča »native« UI vmesnik za ustvarjanje lastnega izgleda aplikacije.

3.6 Framework 7

Framework 7 je prosto in odprto-kodno HTML5 mobilno ogrodje za razvoj hibridnih mobilnih aplikacij za mobilno platformo iOS. Ogorodje je v glavnem namenjeno razvoju iOS aplikacij, vendar v omejenem obsegu podpira tudi razvoj Android aplikacij. Ogorodje Framework 7 razvijalcu ponuja uporabo spletnih tehnologij za razvoj iOS in Android aplikacij. Ogorodje ni kompatibilno z vsemi mobilnimi platformami in je v veliki večini osredotočeno samo na platformo iOS. Poleg tega je v manjšem obsegu osredotočeno tudi na platformo Android in Google material design, kar prinaša odlično uporabniško izkušnjo in preprostost uporabe. Ogorodje Framework 7 je izdalo podjetje iDangerous.us, trenutna različica ogrodja pa je različica 1.4.0, izdana 7. decembra 2015. Lastnosti ogrodja Framework 7:

- Zelo preprosto za uporabo (kot izgradnja spletne strani).
- Odprto-kodno in prosto za uporabo tudi v komercialne namene.
- Namenjeno samo platformi iOS.
- Delno podpira tudi platformo Android in Google Material Design.
- Omogoča uporabo več v naprej ustvarjenih UI komponent in gradnikov.
- Omogoča uporabo visoko zmogljivih Animacij in interakcij.
- Omogoča zelo preprosto in hitro oblikovanje po meri.
- Podpira uporabo ogrodja Cordova.
- Je dobra podpora »native« izgledu in občutku.

Poleg zgoraj naštetih lastnosti obstaja še mnogo drugih, s pomočjo katerih je ogrodje ogrodje Framework 7 zelo preprosto in učinkovito za uporabo[15].

Poglavje 4

Predstavitev evalvacijskega modela

V tem poglavju je prikazan pregled orodij in pristopov, ki smo jih uporabili za zbiranje podatkov o mobilnih ogrodjih. Predstavljeni so kriteriji, s pomočjo katerih smo zasnovali naš evalvacijski model. Poleg modela so predstavljena tudi orodja in pristopi, s pomočjo katerih smo zbirali podatke. Na koncu tega poglavja je predstavljeno, iz katerih spletnih virov smo pridobili mnenja razvijalcev, ki so nam služila pri izvedbi primerjav.

4.1 Izbira mobilnih ogrodij

Ker se je v zadnjem času na spletu pojavlja veliko mobilnih ogrodij, je naš prvi korak pri ustvarjanju evalvacijskega modela vključeval izbiro mobilnih ogrodij, nad katerimi smo želeli izvesti primerjavo. Ker je teh ogrodij na spletu ogromno, smo se odločili, da bomo izbrali pet ogrodij, ki so se pojavljala tako v člankih, ki smo jih predstavili v poglavju 2, kot tudi v raznih drugih spletnih virih. Pri izbiri ogrodij iz teh virov smo se osredotočili na ogrodja, ki omogočajo razvoj hibridnih mobilnih aplikacij (npr. Xamarin, Ionic), izpustili pa smo ogrodja, ki služijo za pakiranje in izdajanje mobilnih

aplikacij (npr. Cordova). Naš cilj je bil primerjati ogrodja, ki se trenutno najbolj uporabljajo za razvoj. Ogrodja, ki smo jih izbrali za primerjavo, so opisana v poglavju 3 in so naša iztočna točka za izvedbo primerjav.

4.2 Izbira kriterijev za primerjavo mobilnih ogrodij

Naslednji korak pri ustvarjanju evalvacijskega modela je vključeval izbiro kriterijev. Kriterije smo želeli zasnovati tako, da pokrivajo čim večje število splošnih in tehničnih lastnosti, saj sta obe vrsti lastnosti zelo pomembni za izbiro optimalnega mobilnega ogrodja. Pri izbiri svojih kriterijev smo si pomagali z različnimi spletnimi viri. Glavni viri za izbiro naših kriterijev so bili v 2. poglavju prikazani modeli in članki. Nekatere kriterije, ki so bili prikazani v člankih, opisanih v poglavju 2 (npr. prototipi, testiranja aplikacij), smo izpustili zaradi pomanjkanja časa za učenje uporabe ogrodij in razvoja prototipov. Ostale kriterije pa smo poskusili združiti v skupine (npr. Tehnične lastnosti ...), ki pa smo jih kasneje razdelili na uporabniške trditve, za lažje ocenjevanje. Poleg tega pa smo pregledali tudi različne spletne strani, kjer razvijalci predstavljajo in primerjajo mobilna ogrodja glede na svoje lastne izkušnje. Pri pregledu teh spletnih virov smo dobili vpogled v tipične zahteve za mobilna ogrodja, na podlagi navedenih lastnosti se razvijalci tipično odločajo za uporabo določenega mobilnega ogrodja.

Na podlagi pregledanih lastnosti in spletnih virov smo se odločili, da bomo svoja izbrana mobilna ogrodja primerjali po naslednjih skupinah kriterijev:

- Kompleksnost.
- Zahtevnost učenja.
- Dokumentacija in podpora.
- Stroški.

- Tehnične lastnosti.
- Vizualne lastnosti.

Zgornje skupine nam bodo služile za prikaz dobrih oziroma slabih lastnosti mobilnih ogrodij. V nadaljevanju bomo na kratko opisali posamezne skupine in kaj želimo primerjati s temi skupinami in kriteriji.

4.2.1 Kompleksnost

Kompleksnost je skupina, s katero želimo prikazati, kako je posamezno mobilno ogrodje zahtevno za uporabo. Cena razvoja mobilnih aplikacij je v veliki večini odvisna od kompleksnosti, saj je razvoj s kompleksnejšimi mobilnimi ogrodji dražji.

Kompleksnost smo opredelili kot čas, ki ga potrebujemo za implementacijo aplikacije s pomočjo mobilnega ogrodja. Razvijalec, ki že ima izkušnje z uporabo kompleksnega mobilnega ogrodja, lahko z njim doseže veliko več, kot pa z uporabo enostavneje zgrajenega mobilnega ogrodja. Po drugi strani pa bo neizkušeni razvijalec porabil veliko časa za učenje kompleksno zgrajenega ogrodja, zato je v njegovem primeru bolje uporabljati enostavno zgrajeno mobilno ogrodje.

4.2.2 Zahtevnost učenja

Zahtevnost učenja je skupina, ki opredeljuje, kako zahtevno je učenje mobilnega ogrodja za razvijalca. Pri tem upoštevamo, ali za učenje o ogrodju obstaja veliko primerov in kako dobri so ti primeri oziroma kako hitro se s pomočjo teh primerov lahko naučimo uporabe mobilnega ogrodja. Zahtevnost učenja je odvisna tudi od predhodnega znanja razvijalca, zato je bolje, da ogrodje temelji na splošno znanih in sprejetih konceptih.

4.2.3 Dokumentacija in podpora

Dokumentacija in podpora je skupina, ki opredeljuje kakovost podpore pri učenju in uporabi mobilnega ogrodja. Dokumentacija sama po sebi služi kot pomoč razvijalcem pri učenju in obvladovanju ogrodja. Pomoč dokumentacije ni potrebna samo, kadar razvijalec začne uporabljati mobilno ogrodje, ampak je potrebna tudi, kadar želi učinkovito uporabiti API mobilnega ogrodja in napredne koncepte. Dokumentacija je dobra, kadar vključuje podrobne opise uporabe API-jev in funkcij ogrodja.

Podpora pa se nanaša na dejstvo, kako velike in močne so skupnosti, ki nudijo podporo razvijalcem za mobilno ogrodje, ter kako pogosto se ta mobilna ogrodja nadgrajujejo. Prvi faktor, ki je lahko pomemben pri odločanju za mobilno ogrodje, je, kako močna je skupnost, ki podpira mobilno ogrodje. Če imajo razvijalci težave pri uporabi ogrodja, se lahko z vprašanji obrnejo na skupnost (ponavadi v obliki foruma). Večja in močnejša kot je skupnost, hitreje bodo razvijalci našli odgovore na svoja vprašanja ter pridobili dodatne informacije o samem delovanju ogrodja. Drugi faktor, ki je lahko pomemben pri odločanju, katero ogrodje uporabiti, je, kako pogosto je to ogrodje nadgrajeno. Pogostejše kot so posodobitve, večja verjetnost je, da se bo ogrodje še izboljševalo in izpopolnjevalo. Če ima ogrodje poleg dejstva, da je pogosto posodobljeno, še veliko skupnost, je podpora za to ogrodje zelo dobra.

4.2.4 Stroški

Stroški je skupina, ki opredeljuje višino stroškov določenega ogrodja oziroma ceno njegove uporabe. Če je mobilno ogrodje odprto-kodno in licencirano z MIT-licenco, se lahko uporablja v komercialne namene z zelo malo omejitvami pri distribuciji. To pomeni, da je ogrodje brezplačno za uporabo, vendar mora vsebovati izvirno oznako za avtorske pravice in licenco[26]. Takšna ogrodja so namenjena predvsem posameznikom, ki želijo razvijati svoje mobilne aplikacije in jih objavljati v raznih spletnih trgovinah.

Vendar pa niso vsa mobilna ogrodja brezplačna za uporabo. Mobilna ogrodja, ki so jih razvila večja podjetja (npr. Xamarin in Sencha), imajo visoke cene za uporabo, saj niso namenjena posameznikom ampak podjetjem. Ponavadi so v te cene vključene različne lastnosti in pravice, ki jih lahko podjetja uporabljajo pri razvoju svojih aplikacij. Dejstvo je, da lahko stroški uporabe mobilnega ogrodja zelo vplivajo na odločitev, katero ogrodje uporabiti.

4.2.5 Tehnične lastnosti

Tehnične Lastnosti je skupina, ki opredeljuje kako dobro mobilno ogrodje podpira nekatere tehnične lastnosti naprav (npr. geo-lokacija, Kamera ...). Ta skupina je pomembna predvsem z vidika razvijalca, saj mora ogrodje kar se da optimalno podpirati različne lastnosti mobilnih naprav. Pri tehničnih lastnostih se bomo osredotočili na tri tipe lastnosti. Prvi tip lastnosti je pregled, kako dobro oziroma v kakšni meri lahko razvijalec s pomočjo ogrodja optimizira napisano kodo. Izvedeti želimo kakšna je modularnost, vzdrževanje kode oziroma je napisana koda jedrnata za pregled.

Drug tip lastnosti so lastnosti, ki se osredotočajo na nalagalni čas aplikacije. Tukaj želimo predvsem izvedeti, kako hitro se ogrodje odzove na AJAX klice pri začetnem nalaganju aplikacije ter kako hiter je nalagalni čas znotraj aplikacije.

Tretji tip lastnosti pa se osredotoča na to, kako dobro ogrodje podpira tehnične lastnosti in različne senzorje naprav, kot so: podpora SQLite, podpora geo-lokacije itd.

4.2.6 Vizualne lastnosti

Vizualne lastnosti je skupina, ki opredeljuje, kako optimizirane UI elemente ponuja ogrodje. Ta skupina je pomembna predvsem z uporabniškega vidika, saj mora ogrodje zagotoviti, da se aplikacija na uporabo občuti čim

bolj kot aplikacija in čim manj kot spletna stran. Poleg tega mora ogrodje zagotoviti tudi, da se aplikacija hitro in učinkovito odzove na uporabniške interakcije, kajti, če je odzivnost aplikacije počasna, potem ogrodje slabo podpira vizualne lastnosti.

Tukaj velja omeniti še dve lastnosti in sicer optimizirani UI elementi in podporo različnim ločljivostim na napravah. – Podpora UI elementov je zelo pomembna, če se bo naša aplikacija izvajala na več napravah z različnimi ločljivostmi. UI elementi morajo biti odzivni in prilagodljivi na različne zaslonе različnih mobilnih naprav.

4.3 Sestavljanje uporabniške ankete

Ko smo izbrali ogrodja za primerjavo in kriterije, s katerimi jih bomo primerjali, je sledilo sestavljanje našega evalvacijskega modela, ki je bil osnova za pripravo ankete. Znotraj kriterijev so se pojavile nekatere lastnosti, ki jih je za posamezno ogrodje mogoče opredeliti le na podlagi mnenj in izkušenj dejanskih razvijalcev oziroma uporabnikov ogrodja.

S pomočjo mnenj teh razvijalcev želimo pridobiti boljšo sliko, kako dobro oziroma slabo ocenjevana mobilna ogrodja podpirajo tehnične in tudi druge lastnosti. Da bi pridobili mnenja, smo na podlagi predhodno predstavljenih kriterijev sestavili vprašalnik, v katerem smo za vsak kriterij navedli trditev, anketirance pa prosili, da na 7-stopenjski Likertovi lestvici ocenijo stopnjo strinjanja oziroma nestrinjanja s podano trditvijo.

Uporabili smo naslednje stopnje:

1. Sploh se ne strinjam
2. Ne strinjam se
3. Se delno ne strinjam
4. Sem nevtralen

5. Se delno strinjam
6. Strinjam se
7. Popolnoma se strinjam

Na podlagi predhodno predstavljenih kriterijev in skupin smo sestavili 22 trditev, s pomočjo katerih želimo primerjati čim več lastnosti mobilnih ogrodij.

1. Kompleksnost

- 1 Ogrodje je preprosto za uporabo.

2. Zahtevnost učenja

- 1 Ogrodje podpira hiter razvoj z dobro podporo in razumljivo sintakso.
- 2 Uporaba ogrodja ne zahteva veliko mentalnega napora.
- 3 Ogrodje ima dobre primere in je preprosto za uporabo.

3. Dokumentacija in podpora

- 1 Ogrodje ima močno skupnost.
- 2 Ogrodje je pogosto posodobljeno.
- 3 Ogrodje ima dobro spletno dokumentacijo.

4. Stroški

- 1 Ogrodje ima nizke stroške podpore.

5. Tehnične lastnosti

- 1 Ogrodje zagotavlja dobro modularnost.
- 2 Ogrodje zagotavlja dober čas nalaganja.

- 3 Ogrodje zagotavlja sredstva za zmanjševanje začetnega časa za nalaganje.
- 4 Ogrodje podpira dolgoročno vzdrževanje in jedrnato kodo.
- 5 Ogrodje je lahko uporabljeno v sodelovanju z drugimi tehnologijami.
- 6 Ogrodje ima odlično podporo senzorjev na vseh napravah
- 7 Ogrodje podpira uvajanje na različnih platformah.
- 8 Ogrodje ima dobro podporo geo-lokacije.
- 9 Ogrodje podpira SQLite.
- 10 Ogrodje podpira datotečni sistem in vhod/izhod.

6. Vizualne lastnosti

- 1 Ogrodje hitro reagira na uporabniške interakcije in ima gladke animacije.
- 2 Ogrodje ponuja podporo za »native« izgled in občutek.
- 3 UI elementi ogrodja so dobro zgrajeni in optimizirani.
- 4 Ogrodje ima dobro podporo ločljivosti na vseh napravah.

4.4 Izvedba ankete

Da bi z anketo pridobili verodostojne odgovore, smo jo objavili na spletnih forumih uradnih spletnih strani mobilnih ogrodij. Žal smo kmalu ugotovili, da bo težko pritegniti zanimanje razvijalcev, saj so le-ti redko pripravljeni odgovarjati na tovrstne ankete.

Zato smo razvijalcem v zameno za njihove odgovore na našo anketo ponudili vpogled v rezultate ankete ter v kasnejšo analizo vseh pridobljenih odgovorov. Vendar je bilo zbiranje podatkov na spletnih forumih vseeno zelo dolgotrajno, saj nismo uspeli pridobiti veliko mnenj razvijalcev. Zato smo se

začeli osredotočati na konkretne razvijalce, zlasti na tiste, kjer smo zasledili, da imajo znanje o uporabi v poglavju 3 navedenih mobilnih ogrodiv. Njihove kontaktne podatke smo pridobili v različnih spletnih trgovinah, kot so: Google Play, App Store itd. – Odziv na uporabniško anketo, izvajano na takšen način, je bil precej boljši.

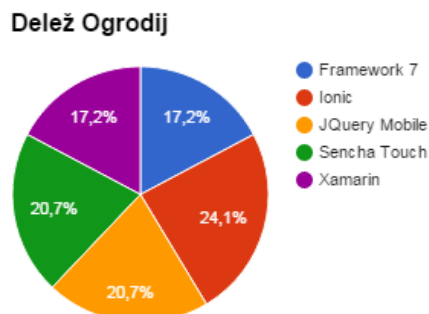
Poglavje 5

Predstavitev rezultatov modela

V tem poglavju bomo predstavili rezultate našega modela, ki je bil osnovan na mnenjih razvijalcev in na podatkih, ki smo jih pridobili s strani proizvajalcev in spletnih skupnosti. Najprej bomo predstavili vzorec, na katerem je bila opravljena analiza, nato pa bomo predstavili rezultate, ki smo jih pridobili s pomočjo uporabniške ankete, ter podrobnosti, kaj ti podatki predstavljajo. Na koncu bomo podatke, ki smo jih pridobili na strani proizvajalcev in mnenja posameznih razvijalcev, prikazali v skupni tabeli. Tako bomo imeli vpogled v vse lastnosti mobilnih ogrodij in bomo videli, katera mobilna ogrodja imajo boljše lastnosti kot druga.

5.1 Rezultati ankete

Z anketo, opisano v prejšnjem poglavju, smo iz različnih spletnih virov, forumov in skupnosti pridobili mnenja razvijalcev. Razvijalci so ocenjevali stopnjo strinjanja oziroma nestrinjanja s trditvami, ki smo jih opisali v poglavju 4.3. Ker je bilo pridobivanje podatkov zelo težavno, saj veliko razvijalcev ni želelo podati svojega mnenja, smo se odločili, da bomo imeli majhen vzorec, s pomočjo katerega ne bo mogoče sklepati o celotni populaciji. Na sliki 5.1 je prikazan delež odgovorov, ki smo ga pridobili za posamezna mobilna ogrodja.



Slika 5.1: Delež odgovorov za posamezno ogrodje

Kot je razvidno iz slike, smo največji delež odgovorov (24,1 %) prejeli za ogrodje Ionic. Po tem lahko sklepamo, da je to ogrodje eno izmed bolj popularnih mobilnih ogrodij med razvijalci. Po drugi strani lahko sklepamo, da sta ogrodji Xamarin in Framework7 eni izmed manj popularnih ogrodij, saj sta prejeli najnižji delež (17,2 %) odgovorov.

5.1.1 Vzorec

Ker smo se za izvedbo primerjav med mobilnimi ogrodji odločili za izvedbo uporabniške ankete, si je bilo potrebno izbrati vzorec, v katerem želimo izvesti primerjave. **Vzorec** je del populacije, v kateri se tipično izvajajo analize in študije. Na podlagi vzorca lahko sklepamo, da rezultati analiz veljajo za celotno populacijo oziroma za določen krog strokovnjakov. Težave se ponavadi pojavijo pri izbiri vzorca, saj mora biti dovolj velik, da lahko z njim zagotavljamo reprezentativnost podatkov za celotno populacijo [20].

Naš vzorec predstavljajo razvijalci, ki imajo izkušnje z razvojem hibridnih mobilnih aplikacij s pomočjo v poglavju 3 navedenih mobilnih ogrodij. Ker smo si za primerjavo izbrali 5 različnih mobilnih ogrodij, smo želeli pridobiti mnenja najmanj 50 razvijalcev. To pomeni, da smo želeli za vsako mobilno ogrodje pridobiti najmanj 10 mnenj, da smo lahko naredili analizo

med mobilnimi ogrodji

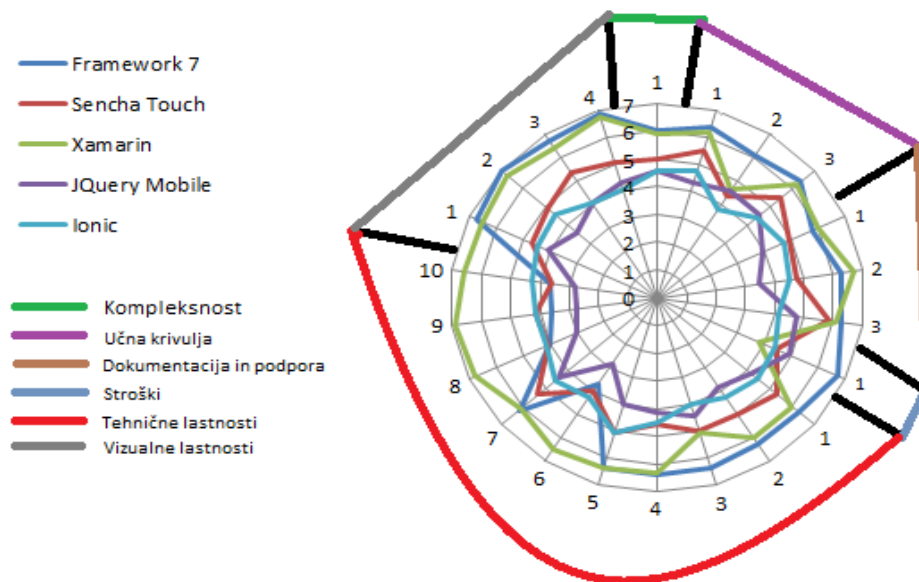
Razlog, da smo se odločili za tako majhen vzorec, leži predvsem v težavi, opisani v poglavju 4.4, saj veliko razvijalcev ni želelo podati svojih mnenj o mobilnih ogrodjih, to pa je bilo časovno problematično. Števila odgovorov, ki smo jih uspeli pridobiti s pomočjo ankete, so naslednja:

- **Ionic** - 14 odgovorov,
- **JQuery Mobile** - 12 odgovorov,
- **Xamarin** - 10 odgovorov,
- **Sencha Touch** - 12 odgovorov in
- **Framework 7** - 10 odgovorov.

5.1.2 Analiza odgovorov

Ocene, ki smo jih pridobili s pomočjo uporabniške ankete, smo strnili v grafe, ki so prikazani na slikah: 5.1, 5.2 in 5.3. Na grafu, prikazanem na sliki 5.2, so opredeljene aritmetične sredine ocenjenih karakteristik za vsako posamezno ogrodje posebej tako, da smo zaradi preglednosti označili, katere trditve spadajo k posameznim kriterijem, ki so opisani v poglavju 4.2. – V omenjenem poglavju so obrazloženi tudi pomeni posameznih števil, ki so na diagramu, prikazanem na sliki 5.2.

Kot je razvidno na grafu, najbolj izstopajo ocene Tehničnih in Vizualnih lastnosti mobilnih ogrodij Xamarin in Framework 7. Razvijalci imajo glede Vizualnih lastnosti omenjenih mobilnih ogrodij precej boljše mnenja, kot pa o ostalih mobilnih ogrodjih. Poleg tega je na grafu tudi zelo dobro razvidno izstopanje Tehničnih lastnosti ogrodja Xamarin. Glede na mnenja razvijalcev, ima ogrodje Xamarin zelo dobro podporo v vseh tehničnih lastnostih. Po drugi strani pa ima ogrodje Framework 7 glede na ocene boljše podporo pri času nalaganja in pri optimizaciji programske kode ter slabšo podporo

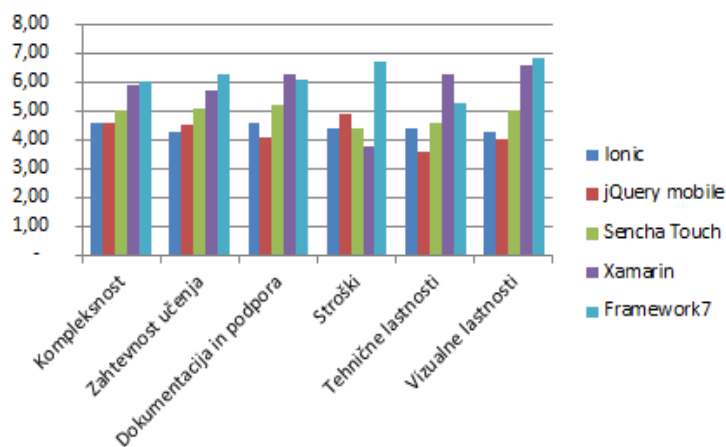


Slika 5.2: Poprečja posameznih ogrodij

senzorjev. Ostala mobilna ogrodja imajo na celotnem grafu približno enakomerne ocene posameznih trditev, z izjemo manjših anomalij.

Poleg vidnih izstopanj pri Tehničnih in Vizualnih lastnostih omenjeni ogrodji Framework 7 in Xamarin izstopata z zelo dobro oceno za dokumentacijo in podporo ter za enostavnost uporabe. Prav tako ogrodje Framework 7 izstopa tudi po ceni in po učni krivulji, saj sta bila oba kriterija ocenjena zelo dobro (v primerjavi z drugimi mobilnimi ogrodji).

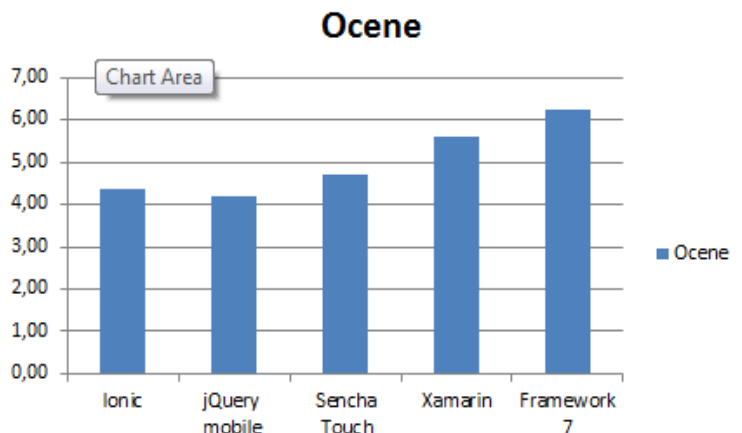
Graf 5.3 prikazuje poprečne ocene skupin kriterijev. Pri ceni najbolj izstopa ogrodje Xamarin, kjer večina vprašanih meni, da ima ogrodje zelo visoke stroške pri uporabi in podpori. Druga točka, ki je tudi dobro vidna in jo je vredno omeniti, so Tehnične lastnosti ogrodja Framework 7. Glede na ocene ostalih kriterijev tega ogrodja, so razvijaci Tehnične lastnosti ocenili slabše, po čemer lahko sklepamo, da se le-ta v praksi ne izkažejo tako dobro. Ostala mobilna ogrodja so ocenjena zelo enakomerno, čeprav se pri stroških in Tehničnih lastnostih ogrodja Sencha Touch pojavljajo manjša odstopanja.



Slika 5.3: Uporabniške trditve po skupinah kriterijev

Graf, prikazan na sliki 5.3, odlično prikazuje ocene, ki smo jih dobili z združitvijo kriterijev po njihovih skupinah. Njegov namen je prikazati primerjavo skupin kriterijev za mobilna ogrodja. Graf, prikazan na sliki 5.4, prikazuje skupne ocene mobilnih ogrodij, ki smo jih pridobili z izračunom poprečja kriterijev posameznih mobilnih ogrodij.

Kot je razvidno z grafa, so razvijalci najbolj ocenili ogrodje Framework 7, saj se njegova poprečna ocena giblje med 6 in 7. Ogrodje, ki so ga razvijalci ocenili najslabše, je bilo ogrodje jQuery Mobile. Njegova ocena se giblje malo nad oceno 4. V splošnem lahko rečemo, da sta ogrodji Xamarin in Framework 7 ocenjeni zelo dobro, medtem ko so ogrodja Ionic, jQuery mobile in Sencha Touch ocenjena slabše.



Slika 5.4: Končne ocene ogrodij

5.2 Podatki, pridobljeni s strani proizvajalcev in skupnosti

V prejšnjem poglavju smo predstavili rezultate, ki smo jih pridobili s pomočjo mnenj razvijalcev. Za kriterij, kot so: Tehnične lastnosti, smo se oprli na mnenja razvijalcev. Za ostale kriterije, kot sta: Cena in Dokumentacija, pa smo podatke poiskali na spletnih straneh različnih skupnosti, forumov in proizvajalcev mobilnih ogrodij. Le-ti so predstavljeni v tem poglavju.

5.2.1 Podpora različnim platformam

Mobilna ogrodja podpirajo različne mobilne platforme, vendar pa velika večina mobilnih ogrodij ne podpira čisto vseh platform, ki so na trgu. (jQuery mobile je eno izmed redkih ogrodij, ki podpira vse mobilne platforme.) V tem poglavju smo predstavili, katere mobilne platforme podpirajo posamezna

mobilna ogrodja.

	Ionic	jQuery mobile	Sencha Touch	Xamarin	Framework 7
Android	Podpira	Podpira	Podpira	Podpira	Delno Podpira
iOS	Podpira	Podpira	Podpira	Podpira	Podpira
Windows Phone	Ne Podpira	Podpira	Podpira	Podpira	Ne Podpira
Kindle	Ne Podpira	Podpira	Podpira	Ne Podpira	Ne Podpira
Blackberry	Podpira	Podpira	Podpira	Ne Podpira	Ne Podpira
Bada	Ne Podpira	Podpira	Ne Podpira	Ne Podpira	Ne Podpira
Tizen	Ne Podpira	Podpira	Podpira	Ne Podpira	Ne Podpira

Tabela 5.1: Podpora različnih platform

Kot je razvidno iz tabele 5.1, mobilno ogrodje Ionic podpira platforme Android, iOS in Blackberry, vendar se lahko to v prihodnosti spremeni, saj je ogrodje uradno še v beta fazi razvoja. Ogrodje Xamarin nudi podporo samo platformama Android in iOS s pomočjo knjižnic Xamarin.Android in Xamarin.iOS. Poleg tega ogrodje Xamarin posredno prek orodja Visual Studio podpira tudi platformo Windows Phone. Ogrodje Framework 7 v celoti podpira platformo iOS, delno podporo pa nudi tudi platformi Android. To pomeni, da lahko razvijalci za razvoj Android aplikacij uporabljajo samo nekatere knjižnice, ki to omogočajo. Ogrodje Sencha Touch, podpira vse mobilne platforme, razen platforme Bada, saj to mobilno platformo počasi zamenjuje platforma Tizen, za katero pa ogrodje nudi polno podporo. Ogrodje, ki podpira vse mobilne platforme, je ogrodje jQuery mobile.

5.2.2 Splošni podatki s strani proizvajalcev

V tem poglavju se bomo osredotočili na podatke, ki smo jih pridobili z raznih spletnih virov, kot so: forumi, skupnosti in spletne strani proizvajalcev mobilnih ogrodij. S temi podatki želimo se dodatno podkrepiti mnenja razvijalcev in podati priporočilo, katero ogrodje uporabiti.

Stroški

Cene podpore posameznih mobilnih ogrodij, ki smo jih našli na uradnih spletnih straneh proizvajalcev, so prikazane v tabeli 5.2. Prikazane cene v tabeli so skladne z mnenji razvijalcev, ki smo jih predstavili v poglavju 5.1.2.

Ogrodje	Cena
Ionic	0 \$
jQuery Mobile	0 \$
Sencha Touch	1995 \$/leto
Xamarin	25-1889 \$/leto
Framework 7	0 \$

Tabela 5.2: Cene posameznih ogrodij

Ogrodja jQuery Mobile, Framework 7 in Ionic so licencirana z MIT licenco, kar pomeni, da lahko razvijalci ta ogrodja uporabljajo brezplačno. To velja tudi za uporabo mobilnih ogrodij v komercialne namene, saj mora programska oprema, razvita s temi ogrodji, vsebovati pogoje uporabe in znak za avtorske pravice[26]. Ogrodji Xamarin in Sencha Touch sta zaradi svoje visoke cene namenjeni podjetjem, vendar pa poleg tega proizvajalci ponujajo tudi preizkusno verzijo za posamezne razvijalce. Glede na ceno, prikazano v tabeli 5.2, sta ogrodji namenjeni podjetjem z več kot 5 razvijalci in za takšno ceno ponujata dodatne lastnosti in pravice, ki jih lahko podjetja uporabljajo v svoje namene.

Tehnične lastnosti

Geo-lokacija je lastnost, ki na mobilni napravi prek GPS sistema sporoča svojo trenutno lokacijo. V tabeli 5.3 je prikazano, kako dobro oziroma slabo mobilna ogrodja podpirajo sistem geo-lokacije.

Ogrodje Xamarin ima odlično podporo za geo-lokacijo, saj ima ogrodje že vgrajeno knjižnico, ki skrbi za hitro in učinkovito delovanje te lastnosti.

Ogrodje	Podpora za geo-lokacijo
Ionic	Dobra
jQuery Mobile	Dobra
Sencha Touch	Dobra
Xamarin	Odlična
Framework 7	Dobra

Tabela 5.3: Podpora za geo-lokacijo

Ostala mobilna ogrodja uporabljajo JavaScript povezavo na geo-lokacijski API (npr. google maps API). Geo-lokacijski API nato poskrbi za to, da naprava komunicira z GPS in pridobiva podatke o trenutni lokaciji. Zato, ker povezava na API poteka z JavaScript tehnologijo, to upočasnjuje delovanje funkcij in zato je podpora za geo-lokacijo pri ostalih mobilnih ogrodjih glede na najdene podatke slabša kot podpora pri ogrodju Xamarin.

Preprostost sintakse predstavlja samo sintakso mobilnega ogrodja in njeno razumljivost. Če je sintaksa mobilnega ogrodja težko razumljiva, bo razvijalec porabil več časa za učenje mobilnega ogrodja. Tabela 5.4 prikazuje, kako preprosto oziroma težavno je razumevanje sintakse posameznega mobilnega ogrodja.

Ogrodje	Preprostost sintakse
Ionic	Preprosta sintaksa
jQuery Mobile	Preprosta sintaksa
Sencha Touch	Težko razumljiva sintaksa
Xamarin	Preprosta sintaksa
Framework 7	Preprosta sintaksa

Tabela 5.4: Preprostost sintakse

Mobilno ogrodje Sencha Touch ima za začetnika zelo nerazumljivo sintakso. Kot smo opisali v poglavju Zahtevnost učenja, je sintaksa ogrodja primerljiva z učenjem novega programskega jezika, zato lahko rečemo, da ogrodje nima preproste sintakse. Ostala mobilna ogrodja imajo relativno preprosto sintakso, saj razvijalci potrebujejo le znanje spletnih tehnologij (Ionic, jQuery mobile, Framework 7) oziroma znanje programskega jezika C# (Xamarin).

Vtičniki so programske komponente, ki dodajo določeno lastnost obstoječemu računalniškemu programu. V našem primeru, želimo izvedeti, kako velik nabor vtičnikov podpirajo posamezna mobilna ogrodja. V tabeli 5.5 je prikazano, kakšen nabor vtičnikov podpirajo posamezna mobilna ogrodja.

Ogrodje	Podpora vtičnikov
Ionic	Zelo malo vtičnikov
jQuery Mobile	Zelo veliko vtičnikov
Sencha Touch	Malo vtičnikov
Xamarin	Ni podatka
Framework 7	Ni podatka

Tabela 5.5: Podpora vtičnikov

Kot je razvidno iz tabele 5.5, ima ogrodje Ionic zelo malo število vtičnikov, kar je posledica tega, da se ogrodje še razvija. Ogrodje Sencha Touch ima (za razliko od ogrodja Ionic) malo večje število vtičnikov, kar pomeni, da lahko v aplikacije, zgrajene s pomočjo tega ogrodja, dodamo nekoliko več vtičnikov kot pri ogrodju Ionic. Največ vtičnikov pa ima zagotovo ogrodje jQuery mobile, ki ima veliko število vtičnikov, ki jih lahko dodamo v mobilno aplikacijo. Za ogrodji Xamarin in Framework 7 nismo našli o številu vtičnikov nobenih podatkov.

Dokumentacija in podpora

Dokumentacija je za razvijalce zelo pomemben vir informacij o samem mobilnem ogrodju. Nudi vpogled v funkcije in lastnosti mobilnega ogrodja ter opredeljuje primere uporabe mobilnega ogrodja. V tabeli 5.6 je prikazano, katera ogrodja imajo dobro napisano dokumentacijo z veliko zanimivimi informacijami in uporabnimi primeri.

Ogrodje	Dokumentacija
Ionic	Dobra
jQuery Mobile	Odlična
Sencha Touch	Odlična
Xamarin	Dobra
Framework 7	Dobra

Tabela 5.6: Dokumentacija

Ogrodji jQuery mobile in Sencha Touch imata v svojih dokumentacijah zelo dobro opisano uporabo vseh lastnosti in knjižnic, ki jih razvijalci lahko uporabijo. Poleg tega imata ogrodji na strani, kjer se nahaja dokumentacija, veliko primerov, kako ogrodji uporabljati. Ogrodje Ionic ima zelo kakovostno, dobro opisano dokumentacijo z nekaj manjšimi pomanjkljivostmi. Ogrodji Framework 7 in Xamarin imata kakovostno dokumentacijo z obširnimi opisi nekaterih funkcij in lastnosti, vendar pa je v dokumentaciji navedenih le malo primerov in sta zato manj primerni za razvijalce, ki se šele učijo, kako uporabljati ogrodje.

Skupnost je skupina razvijalcev, ki v obliki spletnih forumov nudijo pomoč in informacije o uporabi mobilnih ogrodij. Razvijalci se lahko na skupnost obrnejo z raznimi vprašanji glede razvoja oziroma težav, povezanih z razvojem. V tabeli 5.7 so prikazane velikosti skupnosti za posamezna mobilna ogrodja.

Ogrodje	Skupnost
Ionic	Majhna - hitro raste
jQuery Mobile	Velika, aktivna
Sencha Touch	Velika
Xamarin	Srednje velika
Framework 7	Majhna - ena platforma

Tabela 5.7: Skupnosti

Ogrodje jQuery mobile ima zelo veliko, močno in vplivno skupnost, saj je zaradi njegove preprostosti se vedno eno izmed najbolj uporabljanih ogrodij. Tudi ogrodje Sencha Touch ima veliko skupnost kljub temu, da nima enostavne učne krivulje. Xamarin ima (v primerjavi z ogrodjema jQuery mobile in Sencha Touch), manjšo skupnost, ki pa je še vseeno velika in dober vir informacij glede uporabe in pomoči za mobilno ogrodje.

Ogrodje Framework 7 ima zelo majhno skupnost, ki pa je posledica tega, da v celoti podpira samo platformo iOS in je večina razvijalcev v tej skupnosti iOS-razvijalcev. Po drugi strani pa skupnost ogrodja Ionic, čeprav je še majhna, iz dneva v dan hitro narašča, saj je ogrodje zaradi svoje preprostosti in uporabnosti čedalje bolj popularno za razvoj.

Vizualne lastnosti

Skladen izgled je lastnost, ki je bolj pomemben z uporabnikovega vidika. To je izgled, ki je skladen z izgledom aplikacij, ki so bile razvite neposredno s programskim jezikom. Hibridna mobilna aplikacija se mora obnašati kot navadna mobilna aplikacija. Tako lahko uporabnik s pomočjo skladnega izgleda izkusi najboljšo uporabniško izkušnjo. V tabeli 5.8 je prikazano, kako dobro mobilna ogrodja podpirajo skladen izgled.

Ogrodje jQuery mobile ima zelo slabo podporo skladnega izgleda, saj apli-

Ogrodje	»Native« izgled in občutek
Ionic	Dobra
jQuery Mobile	Odlična
Sencha Touch	Odlično
Xamarin	Dobro
Framework 7	Dobra

Tabela 5.8: »Native« izgled in občutek

kacije delujejo počasneje kot čiste »native« aplikacije, še posebej na platformi Android. Ionic ima po drugi strani slabo podporo skladnemu izgledu predvsem zato, ker ga razvijalci še izboljšujejo, vendar pa ima v primerjavi z jQuery mobile dosti boljšo podporo. Ogrodji Framework 7 in Sencha Touch imata dobro podporo skladnega izgleda predvsem zaradi dejstva, da aplikacije teh dveh ogrodij omogočajo dodajanje vtičnikov, ki podpirajo skladen izgled. Aplikacije, zgrajene s pomočjo ogrodja Xamarin, so napisane v programskem jeziku C#, kar pomeni, da se aplikacije že v osnovi prevedejo v »native« aplikacije in ponujajo najboljši skladen izgled in občutek.

Uporabniška izkušnja predstavlja uporabnikov občutek delovanja mobilne aplikacije, zgrajene s pomočjo mobilnega ogrodja. Občutiti se mora čim bolj kot navadna aplikacija in čim manj kot spletna aplikacija. V tabeli 5.9 so prikazane ocene, ki smo jih pridobili na strani spletnih skupnosti za mobilna ogrodja. Te ocene so razvijalci napisali na podlagi izkušenj z zagotavljanjem uporabniške izkušnje uporabnikom.

Kot je razvidno s tabele, najboljšo uporabniško izkušnjo nudita ogrodji Framework 7 in Sencha Touch. Sencha Touch, čeprav ima težko učno krivuljo, nudi eno izmed najboljših uporabniških izkušenj tudi pri kompleksnejših mobilnih aplikacijah. Enake lastnosti veljajo tudi pri ogrodju Framework 7, ki je v veliki večini osredotočeno samo na platformo iOS. Ogrodje Ionic

Ogrodje	Uporabniška izkušnja
Ionic	Dobra
jQuery Mobile	Slaba
Sencha Touch	Odlična
Xamarin	Slaba
Framework 7	Odlična

Tabela 5.9: Uporabniška izkušnja

nudi dobro uporabniško izkušnjo, ki pa se se izboljšuje in dopolnjuje. Slabšo uporabniško izkušnjo nudita ogrodji Xamarin in jQuery mobile. jQuery mobile je ogrodje, ki je zasnovano za razvoj enostavnih mobilnih aplikacij pri kompleksnejših mobilnih aplikacijah je uporabniška izkušnja slabša. Po drugi strani pa pri ogrodju Xamarin razvijalec potrebuje ogromno časa, da zagotovi dobro uporabniško izkušnjo za uporabnika.

Animacije omogočajo uporabnikom enostavne interakcije z mobilnimi aplikacijami. Mobilna ogrodja morajo zagotoviti, da so animacije in prehodi med stranmi oziroma zasloni v aplikaciji, gladki in hitri. Gladke in hitre animacije posredno izboljšujejo uporabniško izkušnjo, kot jo občutijo uporabniki. Ocene, prikazane v tabeli 5.7, so splošne narave in so osnovane na lastnih izkušnjah razvijalcev. Te ocene smo osnovali na mnenjih in izkušnjah razvijalcev, ki so jih zapisali na strani spletnih skupnosti. Če bi želeli bolj natančne ocene, bi jih morali primerjati z izvedbo testov zmogljivosti.

Kot je razvidno v tabeli 5.10, imajo ogrodja Ionic, Xamarin in jQuery mobile po mnenjih razvijalcev počasne animacije in tranzicije med stranmi. V primeru ogrodja Ionic se animacije izboljšujejo, medtem ko ima ogrodje Xamarin počasne animacije zaradi uporabe »native« funkcij jezika C# in podpore več platform. Ogrodje Framework 7 ponuja hitre in gladke animacije, ki so poleg optimizacije še strojno pospešene za dodatno uporabniško izkušnjo. Sencha Touch ponuja hitre in gladke animacije, ker ima optimizirane

Ogrodje	Animacije
Ionic	Počasne
jQuery Mobile	Počasne
Sencha Touch	Hitre
Xamarin	Počasne
Framework 7	Hitre

Tabela 5.10: Animacije

rane knjižnice za vizualne lastnosti kot so animacije.

5.2.3 Podpora mobilnih brskalnikov

Mobilni brskalniki so zasnovani in optimizirani za uporabo na mobilnih napravah. Mobilne brskalnike smo v svojo raziskavo vključili kot dodatno lastnost, saj jih mora v današnjih časih podpirati vsako mobilno ogrodje. Programska oprema za mobilne brskalnike mora biti majhna in učinkovita, da lahko podpira nizke kapacitete spomina in nizko pasovno širino brezžičnih naprav[23]. V našem primeru želimo predvsem prikazati, koliko mobilnih brskalnikov podpirajo posamezna mobilna ogrodja.

	Ionic	jQuery mobile	Sencha Touch	Xamarin	Framework 7
Chrome	Podpira	Podpira	Podpira	Podpira	Ne podpira
Safari	Podpira	Podpira	Podpira	Podpira	Podpira
Android Browser	Podpira	Podpira	Ne podpira	Podpira	Delno podpira
iOS Browser	Podpira	Podpira	Ne podpira	Podpira	Podpira
Opera Mini	Ne podpira	Podpira	Ne podpira	Podpira	Ne podpira
Firefox Mobile	Podpira	Podpira	Ne podpira	Podpira	Ne podpira
Opera Mobile	Ne podpira	Podpira	Ne podpira	Podpira	Ne podpira
IE 11	Ne podpira	Podpira	Podpira	Podpira	Ne podpira
Windows Phone Browser	Podpira	Podpira	Podpira	Podpira	Ne podpira

Tabela 5.11: Podpora različnih platform

V tabeli 5.11 je prikazano, katere mobilne brskalnike podpirajo posamezna mobilna ogrodja. Kot je razvidno na tabeli, največ spletnih brskalnikov

podpirata ogrodji Xamarin in jQuery mobile, saj sta ti dve ogrodji prilagojeni za uporabo na različnih napravah. Ogrodje Sencha Touch po drugi strani podpira brskalnike Safari (iOS) in Chrome(Android) ter privzete Windows Phone brskalnike. Najmanj brskalnikov pa podpirata ogrodji Ionic in Framework 7. Ionic ogrodje podpira tako malo število brskalnikov, ker je novo na trgu – medtem ko ogrodje Framework 7 podpira samo brskalnike, ki jih najdemo na platformi iOS. Poleg tega ogrodje Framework 7 nudi tudi delno podporo za privzeti Android brskalnik, ker ogrodje delno podpira platformo Android.

5.2.4 Različice

Različice smo dodali kot še en dodaten kriterij, pri katerem želimo pogledati, kako pogoste so bile izdaje različic posameznih mobilnih ogrodij in kdaj so proizvajalci izdali prvo in zadnjo različico posameznih mobilnih ogrodij. Ti podatki so prikazani v tabeli 5.12.

	Zadnja različica	Datum izdaje prve različice	Datum izdaje zadnje različice	Pogostost izdaje novih različic
Ionic	1.2.1	12. 05. 2015	17. 12. 2015	1 - 2 meseca
jQuery mobile	1.4.5	19. 03. 2013	31. 10. 2014	1 - 3 mesece
Sencha Touch	2.4.2	06. 03. 2012	15. 06. 2015	3 - 10 mesecev
Xamarin	6.0	16. 05. 2011	10. 12. 2015	3 tedne - 2 meseca
Framework 7	1.4.0	06. 02. 2015	07. 12. 2015	2 tedna - 1 mesec

Tabela 5.12: Prikaz podatkov o različicah

Iz tabele je razvidno, da so izdaje ogrodja Framework 7 v povprečju zelo pogoste, saj proizvajalci novo različico ogrodja izdajo nekje med 2 tedni in 1 mesecem. Po drugi strani pa so izdaje novih različic ogrodja Sencha Touch zelo redke, saj jih proizvajalci v povprečju izdajo nekje med 3 in 10 meseci. Ostala mobilna ogrodja imajo (glede na podatke) relativno pogoste izdaje novih različic, saj jih v povprečju izdajajo nekje med 1 in 3 meseci.

Poleg pogostosti izdaj mobilnih ogrodij sta zanimiva podatka tudi datuma izdaje prve in zadnje različice. Nadgrajevanje ogrodij Sencha Touch in jQuery mobile se je upočasnilo oziroma ustavilo. To se odraža v dejstvu, da je od

datuma izdaje zadnje različice do danes minilo več kot 6 mesecev, po čemer lahko sklepamo, da se je izdajanje različic za ta ogrodja upočasnilo. Ogrodji, ki se glede na podatke v tabeli zelo aktivno nadgrajujeta, sta ogrodji Ionic in Framework 7, saj je od datuma zadnje izdaje različic teh ogrodij minil le 1 mesec. Po tem lahko sklepamo, da proizvajalci ti ogrodji zelo aktivno nadgrajujejo in bodo kmalu izdali nove različice.

5.3 Sinteza rezultatov

V prejšnjih poglavjih smo se osredotočili na pregled in analizo ocen, ki so jih podali razvijalci ter na pregled podatkov, ki smo jih pridobili s strani proizvajalcev in spletnih skupnosti. V tem poglavju je prikazan skupen pregled vseh podatkov s strani spletnih skupnosti in proizvajalcev ter ocen, ki so jih zapisali razvijalci. Vse podatke, ki smo jih pridobili s pomočjo mnenj in s strani proizvajalcev ter skupnosti, smo združili v dve tabeli. Dodali pa smo še tri dodatne lastnosti, ki smo jih našli na straneh proizvajalcev in lahko razvijalcem prav tako koristijo pri izbiri najbolj optimalnega mobilnega ogrodja. Te lastnosti so bolj podrobno predstavljene v poglavjih 5.2.1, 5.2.3 in 5.2.4, tukaj pa želimo prikazati samo okvirne podatke, kot so:

- Število platform, ki jih podpira posamezno ogrodje.
- Število spletnih brskalnikov, ki jih podpira posamezno ogrodje.
- Pogostost izdaj mobilnih ogrodij.

V tabeli 5.13 so prikazane ocene, ki smo jih pridobili z izračunom poprečja uporabniških trditev po posameznih kriterijih, opisanih v poglavju 4.2. Te ocene so grafično predstavljene tudi na sliki 5.3. Ocene, prikazane v tabeli 5.13, se raztezajo od 1 (zelo slabo) do 7 (zelo dobro).

V tabeli 5.14 so prikazani vsi podatki, ki smo jih uspeli pridobiti s strani proizvajalcev in spletnih skupnosti mobilnih ogrodij. Ostali podatki, prika-

	Ionic	jQuery mobile	Sencha Touch	Xamarin	Framework 7
Kompleksnost	4.6	4.6	5	5.9	6
Zahtevnost učenja	4.3	4.5	5.1	5.7	6.3
Dokumentacija in podpora	4.6	4.1	5.2	6.3	6.1
Stroški	4.4	4.9	4.4	3.8	6.7
Tehnične lastnosti	4.4	3.6	4.6	6.3	5.3
Vizualne lastnosti	4.3	4	5	6.6	6.8

Tabela 5.13: Prikaz ocen kriterijev

zani v tabeli 5.14, pa so povzeti s tabel, prikazanih v poglavju 5.2.2. Tabela podaja celovit vpogled v lastnosti vseh mobilnih ogrodij.

	Ionic	jQuery mobile	Sencha Touch	Xamarin	Framework 7
Število platform	3	7	6	3	1.5
Število mobilnih brskalnikov	6	9	4	9	2.5
Izdaje Različic	1 - 2 meseca	1 - 3 mesece	3 - 10 mesecev	1 mesec	2 tedna - 1 mesec
Cena	0	0	1995 \$	25 \$ - 1899 \$	0
Podpora Geolokaciji	Dobra	Dobra	Dobra	Odlična	Dobra
Native izgled in občutek	Slaba podpora	Zelo slaba podpora	Dobra podpora	Odlična podpora	Dobra podpora
Dokumentacija	Dobra	Odlična	Odlična	Dobra	Dobra
Skupnost	Majhna - hitro raste	Velika aktivna	Velika	Srednje velika	Majhna - ena platforma
Uporabniška izkušnja	Dobra	Slaba	Odlična	Slaba	Odlična
Animacije	Počasne	Počasne	Hitre	Počasne	Hitre
Preprostost sintakse	Preprosta sintaksa	Preprosta sintaksa	Težko razumljiva sintaksa	Preprosta sintaksa	Preprosta sintaksa
Podpora vtičnikov	Zelo majhna podpora	Velika podpora	Majhna podpora	N/A	N/A

Tabela 5.14: Prikaz podatkov proizvajalcev

Na podlagi zgoraj opravljene analize podajamo naslednja priporočila:

- V primeru, da razvijalci iščejo mobilno ogrodje z nizko ceno uporabe, priporočamo uporabo ogrodij: Ionic, Framework 7 in jQuery mobile, saj so prosta za uporabo v primerjavi z ogrodjema Xamarin in Sencha Touch.
- V primeru, da razvijalci želijo ogrodje, ki je enostavno za uporabo, priporočamo uporabo ogrodij jQuery mobile in Framework 7, saj za njuno uporabo razvijalci potrebujejo osnovna znanja spletnih tehnologij.
- V primeru, da razvijalci iščejo mobilno ogrodje, ki ima preprosto učno krivuljo in ima dobro napisano dokumentacijo ter močno podporo sku-

pnosti, priporočamo uporabo ogrodja jQuery mobile. Ogrodje ima odlično napisano dokumentacijo z veliko primeri, veliko močno in zelo aktivno skupnost, za njegovo uporabo razvijalci potrebujejo le znanje knjižnice jQuery.

- V primeru, da razvijalci pri ogrojdih iščejo dobro podporo tehničnih lastnosti, jim priporočamo uporabo ogrodja Xamarin, saj odlično podpira vse tehnične lastnosti in je zelo robustno pri razvoju kompleksnejših mobilnih aplikacij.
- V primeru, da razvijalci pri ogrojdih iščejo dobro podporo Vizualnih lastnosti, priporočamo uporabo ogrodijs Sencha Touch in Framework 7. Ogrodji zagotavljata hitre animacije, odlično uporabniško izkušnjo ter dobro podporo za skladen izgled.

Zgoraj navedena priporočila so osnovana na podatkih, ki smo jih pridobili tako na strani razvijalcev kot tudi na straneh proizvajalcev in spletnih skupnosti.

Poglavje 6

Zaključek

V okviru diplomske naloge smo predstavili model, ki primerja ogrodja za razvoj mobilnih aplikacij. Prav tako smo prikazali podatke o lastnostih posameznih mobilnih ogrodij, ki se v praksi najpogosteje uporabljajo.

V uvodu smo predstavili motiv za naše delo, ki izhaja iz dejstva, da imajo razvijalci mobilnih aplikacij na voljo vrsto ogrodij in zato le težko izberejo najbolj ustrezno ogrodje za svoje potrebe. V drugem poglavju smo predstavili podobne modele, ki smo jih našli v raznih člankih, publikacijah in revijah. Predstavili smo njihove dobre in slabe lastnosti, po katerih smo se poskusili čim bolj ravnati oziroma jih izboljšati. Tretje poglavje je bilo namenjeno kratkemu pregledu in zgodovini posameznih mobilnih ogrodij, ki smo jih izbrali za primerjavo. V četrtem poglavju smo predstavili svoj evalvacijski model in orodja, s pomočjo katerih smo oblikovali svoj model. Peto poglavje je bilo namenjeno predstavitvi rezultatov modela in navajanju priporočil, katero ogrodje je najbolj optimalno za uporabo v določeni situaciji.

Pri pripravi evalvacijskega modela smo naleteli na kar nekaj težav. Glavna težava je bilo pridobivanje mnenj razvijalcev s pomočjo ankete, saj razvijalci predvsem na spletnih forumih niso izrazili zanimanja za navajanje in izražanje svojih mnenj o ogrodjih. Zbiranje podatkov na takšen način nam je vzelo ogromno časa. Vzorec, ki smo ga uspeli pridobiti, sicer ni idealen, kljub

temu pa nam je dal zanimive vpoglede v mnenja razvijalcev o karakteristikah posameznih ogrodiv. Na težave smo naleteli tudi pri zbiranju podatkov o ogrodivih na spletnih straneh, saj nekaterih podatkov na strani proizvajalcev sploh nismo našli. Nekaterne podatke smo zato morali pridobiti iz drugih virov, kot npr.: s strani spletnih skupnosti in forumov.

Čeprav smo v modelu skušali zajeti čim večje število kriterijev, pa je seveda mogoče dodati tudi nove, ki bi bili zanimivi zlasti v primerih posebnih potreb razvijalcev (npr. pri osredotočenosti na razvoj aplikacij, ki uporabljajo kamero itd.). Druga razširitev, ki bi jo lahko naredili na svojem modelu, je dodajanje uteži. V našem modelu predpostavljamo, da so vsi kriteriji enakovredni. S pomočjo uteži pa bi lahko ogrodiva med seboj primerjali po pomembnosti kriterijev, saj bi bolj pomembni kriteriji imeli večjo težo pri odločanju v nasprotju z manj pomembni kriteriji. Na ta način bi uporabniku modela omogočili, da sam določi prioritete posameznih kriterijev.

Literatura

- [1] Kate Dreyer (2015) [Online]. Dosegljivo:
<https://www.comscore.com/Insights/Blog/Mobile-Internet-Usage-Skyrockets-in-Past-4-Years-to-Overtake-Desktop-as-Most-Used-Digital-Platform>. [Dostopano: 17. 9. 2015].
- [2] Top 10 Best Hybrid Mobile App UI Frameworks: HTML5, CSS and JS. [Online]. Dosegljivo:
<http://noeticforce.com/best-hybrid-mobile-app-ui-frameworks-html5-js-css>. [Dostopano: 18. 12. 2015].
- [3] Comparing HTML5 mobile UI frameworks [Online]. Dosegljivo:
<http://noeticforce.com/best-hybrid-mobile-app-ui-frameworks-html5-js-css>. [Dostopano: 19. 12. 2015].
- [4] H. Heitkotter, T. A. Majchrzak, B. Ruland, T. Weber. “Evaluating Frameworks for Creating Mobile Web Apps”, v zborniku: WEBIST 2013 - 9th International Conference on Web Information Systems and Technologies, 2013, str. 209-219.
- [5] R. Vala, R. Jasek. “Performance of Hybrid Mobile Application UI Frameworks”, v zborniku: Applied Mathematics, Computational Science and Engineering , 2014, str. 295-297.
- [6] Hyo-jung Sohn, Min-gyu Lee, Baek-min Seong, Jong-Bae Kim. “Quality Evaluation Criteria Based on Open Source Mobile HTML5 UI Fra-

mework for Development of Cross-Platform ”, v zborniku: International Journal of Software Engineering and Its Applications, 2015, str. 1-10.

- [7] Ionic [Online]. Dosegljivo:
[https://en.wikipedia.org/wiki/Ionic_\(mobile_app_framework\)](https://en.wikipedia.org/wiki/Ionic_(mobile_app_framework)). [Dostopano: 22. 10. 2015].
- [8] Ionic framework [Online]. Dosegljivo:
<http://ionicframework.com/docs>. [Dostopano: 22. 10. 2015].
- [9] JQuery Mobile [Online]. Dosegljivo:
https://en.wikipedia.org/wiki/JQuery_Mobile. [Dostopano: 22. 10. 2015].
- [10] A Touch-Optimized Web Framework. [Online]. Dosegljivo:
<https://jquerymobile.com/> [Dostopano: 22. 10. 2015].
- [11] Sencha Touch [Online]. Dosegljivo:
https://en.wikipedia.org/wiki/Sencha_Touch. [Dostopano: 22. 10. 2015].
- [12] Create native-looking HTML5 apps using JavaScript [Online]. Dosegljivo:
<https://www.sencha.com/products/touch/#overview>. [Dostopano: 22. 10. 2015].
- [13] Xamarin [Online]. Dosegljivo:
<https://en.wikipedia.org/wiki/Xamarin>. [Dostopano: 26. 10. 2015].
- [14] Xamarin Platform [Online]. Dosegljivo:
<https://xamarin.com/platform>. [Dostopano: 26. 10. 2015].
- [15] Framework 7 [Online]. Dosegljivo:
<http://www.idangero.us/framework7/#.Vi6ZZ7cvdQJ>. [Dostopano: 26. 10. 2015].

-
- [16] Top 7 Mobile application HTML5 frameworks [Online]. Dosegljivo:
<http://www.gajotres.net/top-7-mobile-application-html5-frameworks/>.
[Dostopano: 5. 10. 2015].
- [17] Mobile Frameworks. [Online]. Dosegljivo:
<http://www.hongkiat.com/blog/mobile-frameworks/>. [Dostopano: 21.
10. 2015].
- [18] Saul McLeod. Likert Scale [Online]. Dosegljivo:
<http://www.simplypsychology.org/likert-scale.html>, 2008. [Dostopano:
31. 10. 2015].
- [19] Likert scale [Online]. Dosegljivo:
https://en.wikipedia.org/wiki/Likert_scale [Dostopano: 31. 10. 2015].
- [20] Basic statistics: mean, median, average, standard deviation, z-scores,
and p-value. [Online]. Dosegljivo:
[https://controls.engin.umich.edu/wiki/index.php/Basic_statistics:_mean,_median,_average,](https://controls.engin.umich.edu/wiki/index.php/Basic_statistics:_mean,_median,_average,_scores,_and_p-value)
[scores,_and_p-value](https://controls.engin.umich.edu/wiki/index.php/Basic_statistics:_mean,_median,_average,_scores,_and_p-value). [Dostopano: 17.11.2015].
- [21] 3 Pros and Cons to Building Mobile Apps with JQuery Mobile and
HTML5. [Online]. Dosegljivo:
[http://niallohiggins.com/2011/09/23/3-pros-cons-mobile-apps-with-](http://niallohiggins.com/2011/09/23/3-pros-cons-mobile-apps-with-jquery-mobile-and-html5/)
[jquery-mobile-and-html5/](http://niallohiggins.com/2011/09/23/3-pros-cons-mobile-apps-with-jquery-mobile-and-html5/). [Dostopano: 21. 11. 2015].
- [22] Software Versioning. [Online]. Dosegljivo:
https://en.wikipedia.org/wiki/Software_versioning. [Dostopano: 23. 11.
2015].
- [23] Mobile Browser. [Online]. Dosegljivo:
https://en.wikipedia.org/wiki/Mobile_browser. [Dostopano: 23. 11.
2015].

- [24] A. Sommer, S. Krusche “Evaluation of cross-platform frameworks for mobile applications”, v zborniku: Software Engineering (Workshops), 2013, str. 363-375.
- [25] 25 Free Mobile App Development Frameworks! [Online]. Dosegljivo: <http://www.pixelatingbits.com/25-free-mobile-app-development-frameworks/>. [Dostopano: 19. 12. 2015].
- [26] MIT Licene [Online]. Dosegljivo: https://en.wikipedia.org/wiki/MIT_License. [Dostopano: 20. 12. 2015].
- [27] Mobile app [Online]. Dosegljivo: https://en.wikipedia.org/wiki/Mobile_app. [Dostopano: 20. 12. 2015].
- [28] HTML5 in mobile devices [Online]. Dosegljivo: https://en.wikipedia.org/wiki/HTML5_in_mobile_devices. [Dostopano: 20. 12. 2015].